Упражнения 5.10

library(fpp2)

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

## ── Attaching packages ───────────────────────────────────────── fpp2 2.4 ──

## ✓ ggplot2 3.3.3 ✓ fma 2.4   
## ✓ forecast 8.13 ✓ expsmooth 2.3

##

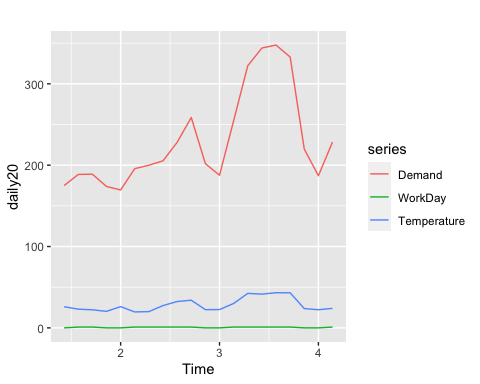
**1. Daily electricity demand for Victoria, Australia, during 2014 is contained in elecdaily. The data for the first 20 days can be obtained as follows.**

daily20 <- head(elecdaily,20)  
daily20

## Time Series:  
## Start = c(1, 4)   
## End = c(4, 2)   
## Frequency = 7   
## Demand WorkDay Temperature  
## 1.428571 174.8963 0 26.0  
## 1.571429 188.5909 1 23.0  
## 1.714286 188.9169 1 22.2  
## 1.857143 173.8142 0 20.3  
## 2.000000 169.5152 0 26.1  
## 2.142857 195.7288 1 19.6  
## 2.285714 199.9029 1 20.0  
## 2.428571 205.3375 1 27.4  
## 2.571429 228.0782 1 32.4  
## 2.714286 258.5984 1 34.0  
## 2.857143 201.7970 0 22.4  
## 3.000000 187.6298 0 22.5  
## 3.142857 254.6636 1 30.0  
## 3.285714 322.2323 1 42.4  
## 3.428571 343.9934 1 41.5  
## 3.571429 347.6376 1 43.2  
## 3.714286 332.9455 1 43.1  
## 3.857143 219.7517 0 23.7  
## 4.000000 186.9816 0 22.3  
## 4.142857 228.4876 1 24.0

*a. Plot the data and find the regression model for Demand with temperature as an explanatory variable. Why is there a positive relationship?*

autoplot(daily20)



model <- tslm(Demand ~ Temperature, data = daily20)  
model

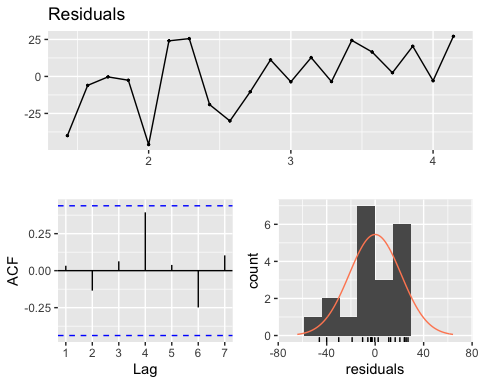
##   
## Call:  
## tslm(formula = Demand ~ Temperature, data = daily20)  
##   
## Coefficients:  
## (Intercept) Temperature   
## 39.212 6.757

Между двумя переменными существует положительная зависимость. Вероятно, по мере повышения температуры включены были кондиционеры, что увеличило энергопотребление.

*b. Produce a residual plot. Is the model adequate? Are there any outliers or influential observations?*

checkresiduals(model$residuals)

## Warning in modeldf.default(object): Could not find appropriate degrees of  
## freedom for this model.

 Нормальная модель, остатки не коррелируют друг с другом. Был outlier.

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

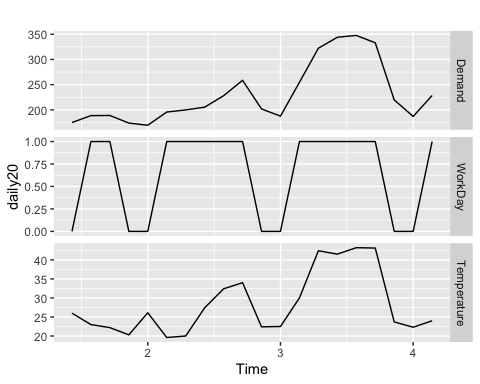
lm\_forecast <- forecast(model,  
 newdata=data.frame(Temperature=c(15,35)))  
lm\_forecast

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 4.285714 140.5701 108.6810 172.4591 90.21166 190.9285  
## 4.428571 275.7146 245.2278 306.2014 227.57056 323.8586

Думаю, модель предсказала правильно, потому что прогнозируемые температуры близки к температурам в данных.

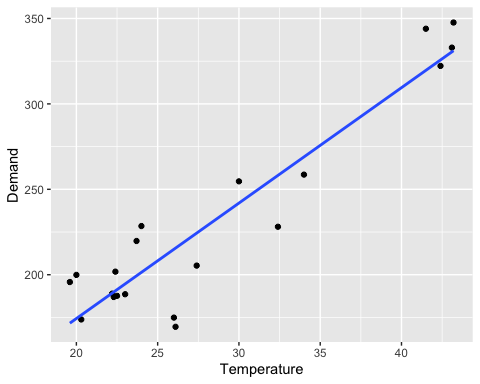
*d. Give prediction intervals for your forecasts. The following R code will get you started:*

autoplot(daily20, facets=TRUE)

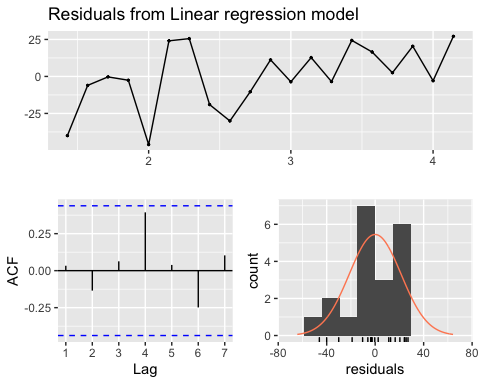


daily20 %>%  
 as.data.frame() %>%  
 ggplot(aes(x=Temperature, y=Demand)) +  
 geom\_point() +  
 geom\_smooth(method="lm", se=FALSE)

## `geom\_smooth()` using formula 'y ~ x'



fit <- tslm(Demand ~ Temperature, data=daily20)  
checkresiduals(fit)



##   
## 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

forecast(fit, newdata=data.frame(Temperature=c(15,35)))

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 4.285714 140.5701 108.6810 172.4591 90.21166 190.9285  
## 4.428571 275.7146 245.2278 306.2014 227.57056 323.8586

80% intervals

lm\_forecast$upper[,1]

## Time Series:  
## Start = c(4, 3)   
## End = c(4, 4)   
## Frequency = 7   
## [1] 172.4591 306.2014

lm\_forecast$lower[,1]

## Time Series:  
## Start = c(4, 3)   
## End = c(4, 4)   
## Frequency = 7   
## [1] 108.6810 245.2278

95% intervals

lm\_forecast$upper[,2]

## Time Series:  
## Start = c(4, 3)   
## End = c(4, 4)   
## Frequency = 7   
## [1] 190.9285 323.8586

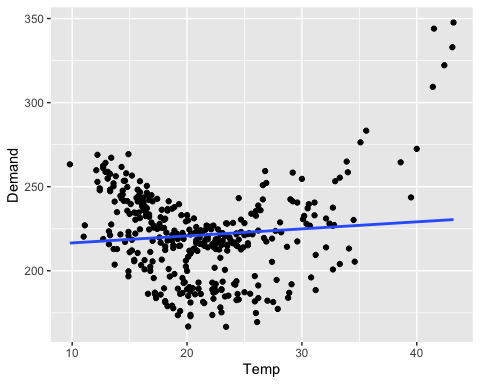
lm\_forecast$lower[,2]

## Time Series:  
## Start = c(4, 3)   
## End = c(4, 4)   
## Frequency = 7   
## [1] 90.21166 227.57056

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

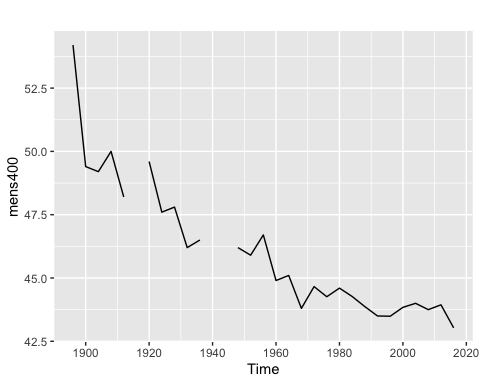
elecdaily %>%  
 as.data.frame() %>%  
 ggplot(aes(x=Temperature, y=Demand)) +  
 ylab("Demand") +  
 xlab("Temp") +  
 geom\_point() +  
 geom\_smooth(method="lm", se=FALSE)

## `geom\_smooth()` using formula 'y ~ x'

 График результатов показывает, что модель некорректно работает на более длительных промежутках времени, она была построена на не слишком большом количестве данных, видно, что простая линейная регрессия недостаточна в этом случае.

**2. Data set mens400 contains the winning times (in seconds) for the men’s 400 meters final in each Olympic Games from 1896 to 2016.** *a. Plot the winning time against the year. Describe the main features of the plot.*

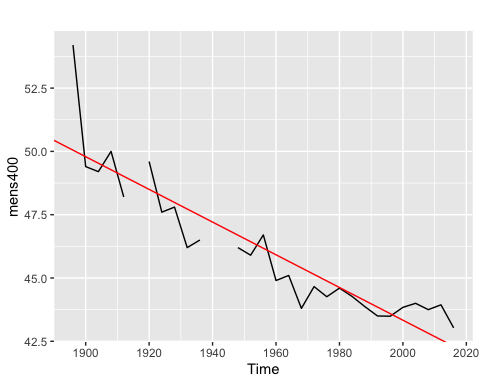
autoplot(mens400)



1. Тренд - рекордное время побед в финале олимпийских соревнований по бегу на 400 м среди мужчин со временем уменьшалось.
2. Есть отсутствующие значения значения.

*b. Fit a regression line to the data. Obviously the winning times have been decreasing, but at what average rate per year? Extract time part from mens400 time series to do linear modeling.*

m4\_time <- time(mens400)  
m4\_tslm <- tslm(mens400 ~ m4\_time, data = mens400)  
autoplot(mens400) +  
 geom\_abline(slope = m4\_tslm$coefficients[2],  
 intercept = m4\_tslm$coefficients[1],  
 colour = "red")

 Коэффициент убывания

m4\_tslm$coefficients[2]

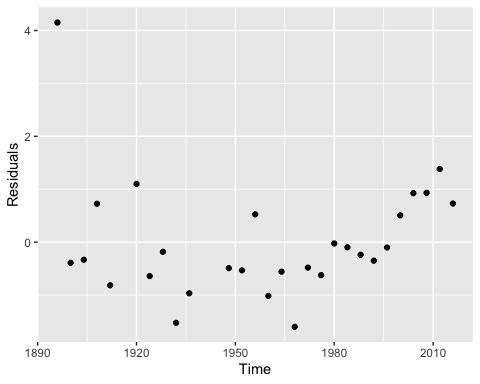
## m4\_time   
## -0.06457385

Уменьшается на 0.06457 секунды в год в среднем.

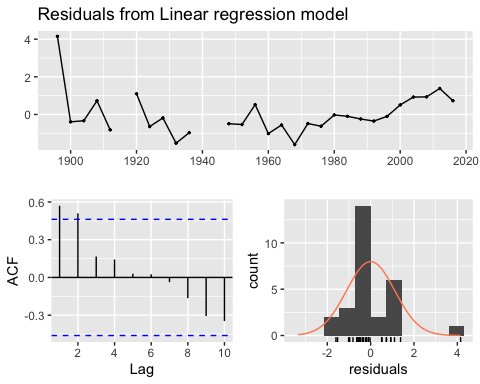
*c. Plot the residuals against the year. What does this indicate about the suitability of the fitted line?*

cbind(Time = m4\_time,   
 Residuals = m4\_tslm$residuals) %>%  
 as.data.frame() %>%  
 ggplot(aes(x = Time, y = Residuals)) +  
 geom\_point() +  
 ylab("Residuals")

## Warning: Removed 3 rows containing missing values (geom\_point).



checkresiduals(m4\_tslm)



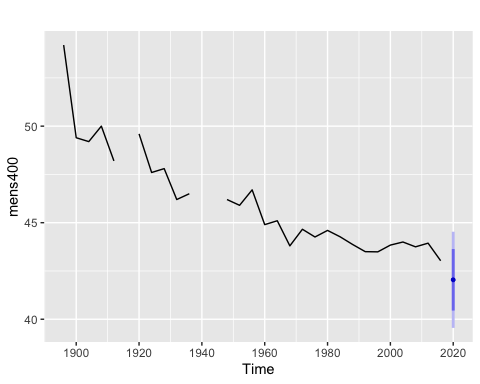
##   
## Breusch-Godfrey test for serial correlation of order up to 6  
##   
## data: Residuals from Linear regression model  
## LM test = 3.6082, df = 6, p-value = 0.7295

График остатков показывает, что модель в целом хорошо соответствует данным для краткосрочных прогнозов.

*d. Predict the winning time for the men’s 400 meters final in the 2020 Olympics. Give a prediction interval for your forecasts. What assumptions have you made in these calculations?*

Построил линейную модель с исключением отсутствующих значений.

m4\_lm <- lm(  
 mens400 ~ m4\_time,   
 data = mens400,  
 na.action = na.exclude  
 )  
  
m4\_forecast <- forecast(  
 m4\_lm,   
 newdata = data.frame(m4\_time = 2020)  
 )  
  
autoplot(mens400) +  
 autolayer(m4\_forecast, PI = TRUE)

 Доверительные интервалы

m4\_forecast$upper

## [,1] [,2]  
## [1,] 43.63487 44.53176

m4\_forecast$lower

## [,1] [,2]  
## [1,] 40.44975 39.55286

80% интервал: 40.45 - 43.63 95% интервал: 39.55 - 44.53

Расчеты были сделаны из предположения, что остатки модели были нормально распределены. Но видно по результату функции checkresiduals, что это не так.

**3. Type easter(ausbeer) and interpret what you see.**

Квартальные анные о производстве пива в Австралии в период Пасхи (Easter), 1- в период была Пасха, 0 - не было, или дробное, если часть квартала пришлось на Пасху. 218 значений, начало - 1-й квартал 1956 г., последнее измерение - 2-й квартал 2010 г.

easter(ausbeer)

## Qtr1 Qtr2 Qtr3 Qtr4  
## 1956 0.67 0.33 0.00 0.00  
## 1957 0.00 1.00 0.00 0.00  
## 1958 0.00 1.00 0.00 0.00  
## 1959 1.00 0.00 0.00 0.00  
## 1960 0.00 1.00 0.00 0.00  
## 1961 0.33 0.67 0.00 0.00  
## 1962 0.00 1.00 0.00 0.00  
## 1963 0.00 1.00 0.00 0.00  
## 1964 1.00 0.00 0.00 0.00  
## 1965 0.00 1.00 0.00 0.00  
## 1966 0.00 1.00 0.00 0.00  
## 1967 1.00 0.00 0.00 0.00  
## 1968 0.00 1.00 0.00 0.00  
## 1969 0.00 1.00 0.00 0.00  
## 1970 1.00 0.00 0.00 0.00  
## 1971 0.00 1.00 0.00 0.00  
## 1972 0.33 0.67 0.00 0.00  
## 1973 0.00 1.00 0.00 0.00  
## 1974 0.00 1.00 0.00 0.00  
## 1975 1.00 0.00 0.00 0.00  
## 1976 0.00 1.00 0.00 0.00  
## 1977 0.00 1.00 0.00 0.00  
## 1978 1.00 0.00 0.00 0.00  
## 1979 0.00 1.00 0.00 0.00  
## 1980 0.00 1.00 0.00 0.00  
## 1981 0.00 1.00 0.00 0.00  
## 1982 0.00 1.00 0.00 0.00  
## 1983 0.00 1.00 0.00 0.00  
## 1984 0.00 1.00 0.00 0.00  
## 1985 0.00 1.00 0.00 0.00  
## 1986 1.00 0.00 0.00 0.00  
## 1987 0.00 1.00 0.00 0.00  
## 1988 0.00 1.00 0.00 0.00  
## 1989 1.00 0.00 0.00 0.00  
## 1990 0.00 1.00 0.00 0.00  
## 1991 1.00 0.00 0.00 0.00  
## 1992 0.00 1.00 0.00 0.00  
## 1993 0.00 1.00 0.00 0.00  
## 1994 0.00 1.00 0.00 0.00  
## 1995 0.00 1.00 0.00 0.00  
## 1996 0.00 1.00 0.00 0.00  
## 1997 1.00 0.00 0.00 0.00  
## 1998 0.00 1.00 0.00 0.00  
## 1999 0.00 1.00 0.00 0.00  
## 2000 0.00 1.00 0.00 0.00  
## 2001 0.00 1.00 0.00 0.00  
## 2002 1.00 0.00 0.00 0.00  
## 2003 0.00 1.00 0.00 0.00  
## 2004 0.00 1.00 0.00 0.00  
## 2005 1.00 0.00 0.00 0.00  
## 2006 0.00 1.00 0.00 0.00  
## 2007 0.00 1.00 0.00 0.00  
## 2008 1.00 0.00 0.00 0.00  
## 2009 0.00 1.00 0.00 0.00  
## 2010 0.00 1.00

**4. An elasticity coefficient is the ratio of the percentage change in the forecast variable (y) to the percentage change in the predictor variable (x). Mathematically, the elasticity is defined as (dy/dx)x(x/y). Consider the log-log model, log y = β0 + β1 logx+ε.Express y as a function of x and show that the coefficient β1 is the elasticity coefficient.**

log(y) = β0 + β1log(x) + ε e^log(y) = e^(β+β1log(x) + ε) y = e^(β + ε) x^c => выразил у как функция от x

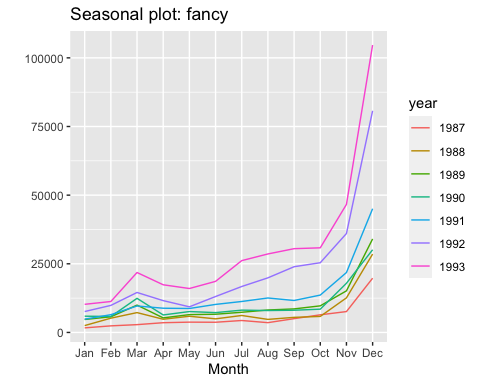
Покажем, что β1 - коэффициент эластичности. log(y) = β0 + β1log(x) + ε β1log(x) = log(y) - β0 - ε β1 = (log(y) - β0 - ε) / log(x)

Таким образом, коэффициент β1 является коэффициентом эластичности, поскольку он показывает отношение процентного изменения прогнозируемой переменной (y) к изменению независимой переменной x. Для каждого изменения в x существует процентное изменение отношения β1 в y.

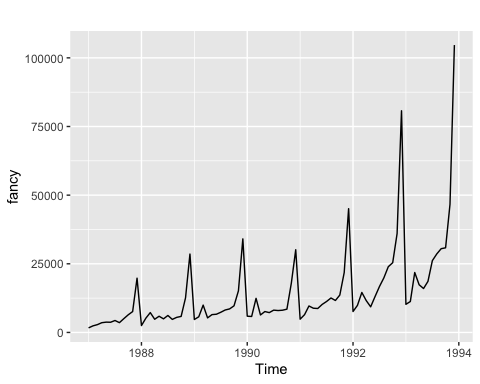
**5. The data set fancy concerns the monthly sales figures of a shop which opened in January 1987 and sells gifts, souvenirs, and novelties. The shop is situated on the wharf at a beach resort town in Queensland, Australia. The sales volume varies with the seasonal population of tourists. There is a large influx of visitors to the town at Christmas and for the local surfing festival, held every March since 1988. Over time, the shop has expanded its premises, range of products, and staff.**

*a. Produce a time plot of the data and describe the patterns in the graph. Identify any unusual or unexpected fluctuations in the time series.*

ggseasonplot(fancy)



autoplot(fancy)

 Продажи в целом увеличивались с января по декабрь. Потом резко росли в декабре. В 1991 году продажи упали в декабре, по сравнения с предыдущими годами увеличения. Еще каждый март также наблюдался рост, но он был меньше, чем рост в декабре.

*b. Explain why it is necessary to take logarithms of these data before fitting a model.* Размер сезонных колебаний должен быть почти одинаковым (стационарным), чтобы хорошо соответствовать модели. Данные показывают, что сезонные колебания увеличиваются по экспоненте. Поэтому необходимо логарифмировать данные. К тому же, если в наборе данных существует закономерность, остатки могут быть неоднородными, их дисперсия можеть быть непостоянной и приводить к проблемам в регрессии.

*c. Use R to fit a regression model to the logarithms of these sales data with a linear trend, seasonal dummies and a “surfing festival” dummy variable.*

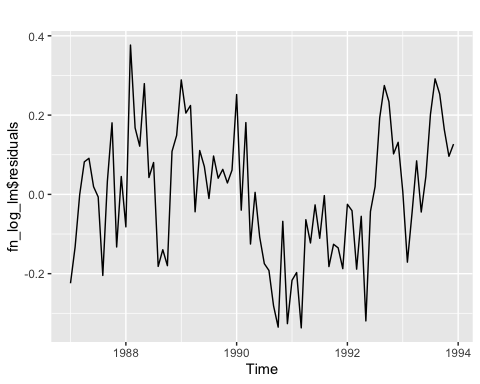
Создадим фиктивную переменную “surfing\_festival”. Значение равно 1, если год > 1988, а месяц - март.

Time <- time(fancy)  
surfing\_festival <- c()  
for(i in 1:length(Time)){  
 month <- round(12\*(Time[i] - floor(Time[i]))) + 1  
 year <- floor(Time[i])  
 if(year >= 1988 & month == 3){  
 surfing\_festival[i] <- 1  
 } else {  
 surfing\_festival[i] <- 0  
 }  
}  
  
fn\_log\_lm <- tslm(  
 BoxCox(fancy, 0) ~ trend + season + surfing\_festival  
 )  
summary(fn\_log\_lm)

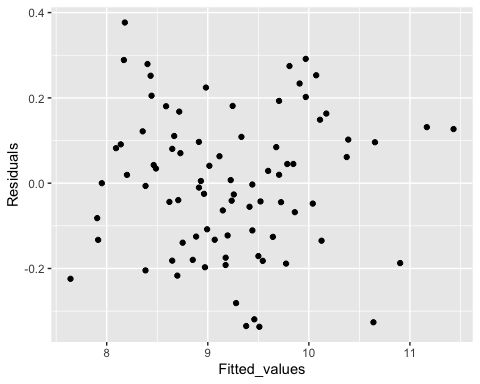
##   
## Call:  
## tslm(formula = BoxCox(fancy, 0) ~ trend + season + surfing\_festival)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.33673 -0.12757 0.00257 0.10911 0.37671   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 7.6196670 0.0742471 102.626 < 2e-16 \*\*\*  
## trend 0.0220198 0.0008268 26.634 < 2e-16 \*\*\*  
## season2 0.2514168 0.0956790 2.628 0.010555 \*   
## season3 0.2660828 0.1934044 1.376 0.173275   
## season4 0.3840535 0.0957075 4.013 0.000148 \*\*\*  
## season5 0.4094870 0.0957325 4.277 5.88e-05 \*\*\*  
## season6 0.4488283 0.0957647 4.687 1.33e-05 \*\*\*  
## season7 0.6104545 0.0958039 6.372 1.71e-08 \*\*\*  
## season8 0.5879644 0.0958503 6.134 4.53e-08 \*\*\*  
## season9 0.6693299 0.0959037 6.979 1.36e-09 \*\*\*  
## season10 0.7473919 0.0959643 7.788 4.48e-11 \*\*\*  
## season11 1.2067479 0.0960319 12.566 < 2e-16 \*\*\*  
## season12 1.9622412 0.0961066 20.417 < 2e-16 \*\*\*  
## surfing\_festival 0.5015151 0.1964273 2.553 0.012856 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.179 on 70 degrees of freedom  
## Multiple R-squared: 0.9567, Adjusted R-squared: 0.9487   
## F-statistic: 119 on 13 and 70 DF, p-value: < 2.2e-16

*d. Plot the residuals against time and against the fitted values. Do these plots reveal any problems with the model?*

autoplot(fn\_log\_lm$residuals)

 Остатки зависят от времени. Это означает, что существует корреляция между остатками и временем.

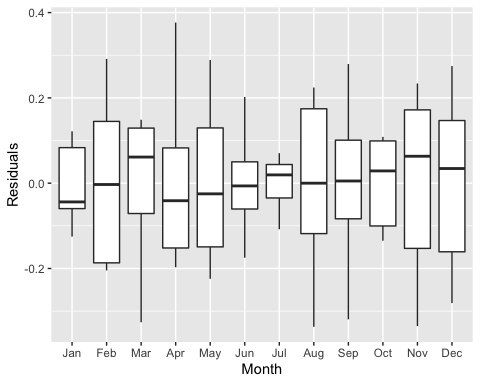
cbind(Residuals = fn\_log\_lm$residuals,  
 Fitted\_values = fn\_log\_lm$fitted.values) %>%  
 as.data.frame() %>%  
 ggplot(aes(x = Fitted\_values,  
 y = Residuals)) +  
 geom\_point()

 Размер остатков изменяется по мере продвижения по оси спрогнозированных значений. Значит, что даже после логарифмического преобразования ошибки все еще остаются неоднородными.

*e. Do boxplots of the residuals for each month. Does this reveal any problems with the model?*

cbind.data.frame(  
 Month = factor(  
 month.abb[round(12\*(Time - floor(Time)) + 1)],  
 labels = month.abb,  
 ordered = TRUE  
 ),  
 Residuals = fn\_log\_lm$residuals  
 ) %>%  
 ggplot(aes(x = Month,  
 y = Residuals)) +  
 geom\_boxplot()

## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.



Несколько месяцев распределение остатков было несимметричным. И в течение некоторых месяцев среднее остатков не равнялась 0. Такие остатки не нормально распределны, что усложняет получение точного интервала прогноза.

*f. What do the values of the coefficients tell you about each variable?*

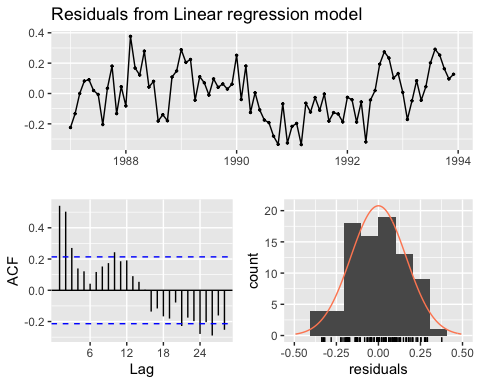
fn\_log\_lm$coefficients

## (Intercept) trend season2 season3   
## 7.61966701 0.02201983 0.25141682 0.26608280   
## season4 season5 season6 season7   
## 0.38405351 0.40948697 0.44882828 0.61045453   
## season8 season9 season10 season11   
## 0.58796443 0.66932985 0.74739195 1.20674790   
## season12 surfing\_festival   
## 1.96224123 0.50151509

Модель имеет положительный тренд (с течением времени объем продаж увеличивается). Все сезонные переменные положительны (объем продаж минимален в январе, а продажи в другие месяцы больше января в течение большего времени). surfing\_festival равен 0.501, тем самым вносит большой вклад в итоговые продажи, то есть, когда проводится фестиваль серфинга, продажи растут.

*g. What does the Breusch-Godfrey test tell you about your model?*

checkresiduals(fn\_log\_lm)

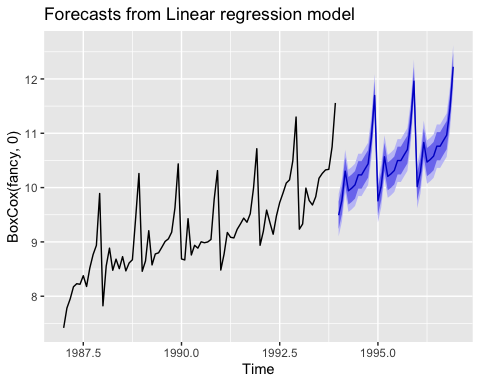


##   
## Breusch-Godfrey test for serial correlation of order up to 17  
##   
## data: Residuals from Linear regression model  
## LM test = 37.954, df = 17, p-value = 0.002494

Значение p-теста меньше 0.05. Это означает, что есть автокорреляция в остатках.

*h. Regardless of your answers to the above questions, use your regression model to predict the monthly sales for 1994, 1995, and 1996. Produce prediction intervals for each of your forecasts.*

#попытка скорректировать модель на мартовсие всплески продаж  
future\_fancy <- rep(0, 36)  
for(i in 1:36){  
 if(i %% 12 == 3){  
 future\_fancy[i] <- 1  
 }  
}  
  
future\_fancy <- ts(data = future\_fancy,  
 start = 1994,  
 end = c(1996, 12),  
 frequency = 12)  
  
  
fn\_log\_lm\_forecast <- forecast(  
 fn\_log\_lm,  
 newdata = data.frame(Time = time(future\_fancy),  
 surfing\_festival = future\_fancy)  
)  
  
autoplot(fn\_log\_lm\_forecast)

 Доверительные интервалы

fn\_log\_lm\_forecast$upper

## [,1] [,2]  
## Jan 1994 9.744183 9.88111  
## Feb 1994 10.017620 10.15455  
## Mar 1994 10.557120 10.69475  
## Apr 1994 10.194296 10.33122  
## May 1994 10.241749 10.37868  
## Jun 1994 10.303110 10.44004  
## Jul 1994 10.486756 10.62368  
## Aug 1994 10.486286 10.62321  
## Sep 1994 10.589671 10.72660  
## Oct 1994 10.689753 10.82668  
## Nov 1994 11.171129 11.30806  
## Dec 1994 11.948642 12.08557  
## Jan 1995 10.011336 10.14984  
## Feb 1995 10.284773 10.42328  
## Mar 1995 10.823938 10.96297  
## Apr 1995 10.461449 10.59996  
## May 1995 10.508903 10.64741  
## Jun 1995 10.570264 10.70877  
## Jul 1995 10.753910 10.89242  
## Aug 1995 10.753440 10.89195  
## Sep 1995 10.856825 10.99533  
## Oct 1995 10.956907 11.09541  
## Nov 1995 11.438282 11.57679  
## Dec 1995 12.215796 12.35430  
## Jan 1996 10.279093 10.41951  
## Feb 1996 10.552530 10.69294  
## Mar 1996 11.091365 11.23212  
## Apr 1996 10.729206 10.86962  
## May 1996 10.776659 10.91707  
## Jun 1996 10.838021 10.97843  
## Jul 1996 11.021667 11.16208  
## Aug 1996 11.021196 11.16161  
## Sep 1996 11.124582 11.26499  
## Oct 1996 11.224664 11.36508  
## Nov 1996 11.706039 11.84645  
## Dec 1996 12.483552 12.62396

fn\_log\_lm\_forecast$lower

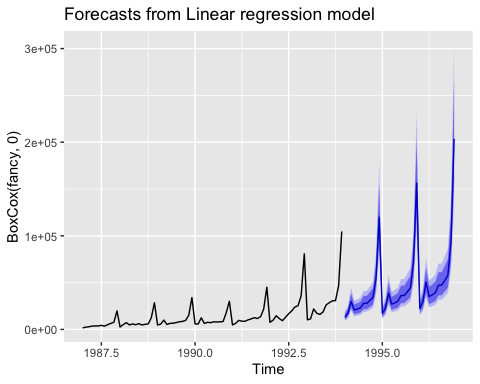
## [,1] [,2]  
## Jan 1994 9.238522 9.101594  
## Feb 1994 9.511959 9.375031  
## Mar 1994 10.048860 9.911228  
## Apr 1994 9.688635 9.551707  
## May 1994 9.736088 9.599161  
## Jun 1994 9.797449 9.660522  
## Jul 1994 9.981095 9.844168  
## Aug 1994 9.980625 9.843698  
## Sep 1994 10.084010 9.947083  
## Oct 1994 10.184092 10.047165  
## Nov 1994 10.665468 10.528541  
## Dec 1994 11.442981 11.306054  
## Jan 1995 9.499844 9.361338  
## Feb 1995 9.773281 9.634775  
## Mar 1995 10.310518 10.171489  
## Apr 1995 9.949957 9.811451  
## May 1995 9.997411 9.858904  
## Jun 1995 10.058772 9.920265  
## Jul 1995 10.242418 10.103911  
## Aug 1995 10.241948 10.103441  
## Sep 1995 10.345333 10.206826  
## Oct 1995 10.445415 10.306908  
## Nov 1995 10.926791 10.788284  
## Dec 1995 11.704304 11.565797  
## Jan 1996 9.760564 9.620151  
## Feb 1996 10.034000 9.893588  
## Mar 1996 10.571566 10.430810  
## Apr 1996 10.210677 10.070264  
## May 1996 10.258130 10.117718  
## Jun 1996 10.319491 10.179079  
## Jul 1996 10.503137 10.362725  
## Aug 1996 10.502667 10.362254  
## Sep 1996 10.606052 10.465640  
## Oct 1996 10.706134 10.565722  
## Nov 1996 11.187510 11.047097  
## Dec 1996 11.965023 11.824611

*i. Transform your predictions and intervals to obtain predictions and intervals for the raw data.*

fn\_log\_lm\_forecast\_transform <- fn\_log\_lm\_forecast  
  
log\_trans <- c('x', 'mean', 'lower', 'upper', 'residuals', 'fitted')  
  
fn\_log\_lm\_forecast\_transform[log\_trans] <- lapply(  
 fn\_log\_lm\_forecast[log\_trans],  
 InvBoxCox,  
 lambda = 0  
)

Перевести log в boxcox

fn\_log\_lm\_forecast\_transform[['model']][['model']][1] <- lapply(  
 fn\_log\_lm\_forecast[['model']][['model']][1],  
 InvBoxCox,  
 lambda = 0  
)  
autoplot(fn\_log\_lm\_forecast\_transform)



Интервалы

fn\_log\_lm\_forecast\_transform$upper

## [,1] [,2]  
## Jan 1994 17054.73 19557.43  
## Feb 1994 22418.00 25707.73  
## Mar 1994 38450.24 44123.68  
## Apr 1994 26750.16 30675.62  
## May 1994 28050.15 32166.37  
## Jun 1994 29825.24 34201.95  
## Jul 1994 35837.72 41096.73  
## Aug 1994 35820.87 41077.41  
## Sep 1994 39722.43 45551.50  
## Oct 1994 43903.67 50346.32  
## Nov 1994 71049.28 81475.41  
## Dec 1994 154607.08 177294.90  
## Jan 1995 22277.59 25587.08  
## Feb 1995 29283.31 33633.55  
## Mar 1995 50208.44 57697.39  
## Apr 1995 34942.16 40133.06  
## May 1995 36640.25 42083.42  
## Jun 1995 38958.95 44746.58  
## Jul 1995 46812.70 53767.06  
## Aug 1995 46790.69 53741.78  
## Sep 1995 51887.06 59595.26  
## Oct 1995 57348.77 65868.34  
## Nov 1995 92807.48 106594.69  
## Dec 1995 201954.08 231955.81  
## Jan 1996 29117.46 33506.86  
## Feb 1996 38274.14 44043.89  
## Mar 1996 65602.25 75517.62  
## Apr 1996 45670.42 52555.15  
## May 1996 47889.88 55109.18  
## Jun 1996 50920.48 58596.65  
## Jul 1996 61185.57 70409.18  
## Aug 1996 61156.80 70376.07  
## Sep 1996 67817.91 78041.33  
## Oct 1996 74956.52 86256.08  
## Nov 1996 121302.09 139588.15  
## Dec 1996 263959.89 303751.35

fn\_log\_lm\_forecast\_transform$lower

## [,1] [,2]  
## Jan 1994 10285.82 8969.583  
## Feb 1994 13520.45 11790.284  
## Mar 1994 23129.40 20155.412  
## Apr 1994 16133.21 14068.696  
## May 1994 16917.24 14752.395  
## Jun 1994 17987.81 15685.969  
## Jul 1994 21613.98 18848.111  
## Aug 1994 21603.82 18839.249  
## Sep 1994 23956.87 20891.193  
## Oct 1994 26478.61 23090.230  
## Nov 1994 42850.31 37366.903  
## Dec 1994 93244.59 81312.400  
## Jan 1995 13357.65 11629.938  
## Feb 1995 17558.28 15287.252  
## Mar 1995 30046.98 26146.972  
## Apr 1995 20951.33 18241.435  
## May 1995 21969.51 19127.918  
## Jun 1995 23359.80 20338.387  
## Jul 1995 28068.91 24438.412  
## Aug 1995 28055.72 24426.922  
## Sep 1995 31111.50 27087.467  
## Oct 1995 34386.35 29938.733  
## Nov 1995 55647.40 48449.831  
## Dec 1995 121091.75 105429.448  
## Jan 1996 17336.40 15065.329  
## Feb 1996 22788.25 19802.984  
## Mar 1996 39009.73 33887.802  
## Apr 1996 27191.96 23629.808  
## May 1996 28513.41 24778.151  
## Jun 1996 30317.82 26346.183  
## Jul 1996 36429.60 31657.322  
## Aug 1996 36412.48 31642.439  
## Sep 1996 40378.47 35088.887  
## Oct 1996 44628.77 38782.394  
## Nov 1996 72222.70 62761.521  
## Dec 1996 157160.50 136572.460

*j. How could you improve these predictions by modifying the model?*

Можно попробовать избавиться от каких-то предикторов, если есть взаимная корреляция. Также можно попробовать добавить смещенные по времени копии переменных (lagged copies)

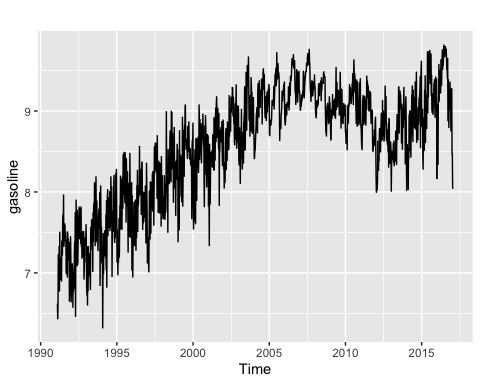
**6. The gasoline series consists of weekly data for supplies of US finished motor gasoline product, from 2 February 1991 to 20 January 2017. The units are in “thousand barrels per day”. Consider only the data to the end of 2004.**

*a. Fit a harmonic regression with trend to the data. Experiment with changing the number Fourier terms. Plot the observed gasoline and fitted values and comment on what you see.*

head(gasoline)

## Time Series:  
## Start = 1991.1   
## End = 1991.19582477755   
## Frequency = 52.1785714285714   
## [1] 6.621 6.433 6.582 7.224 6.875 6.947

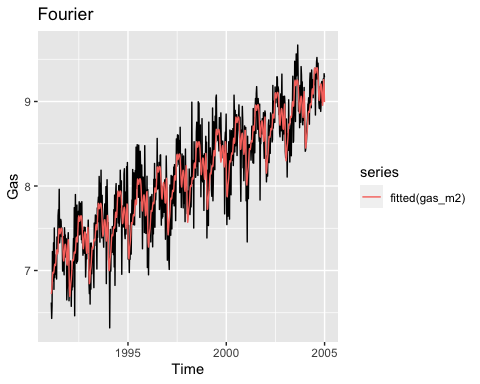
autoplot(gasoline)



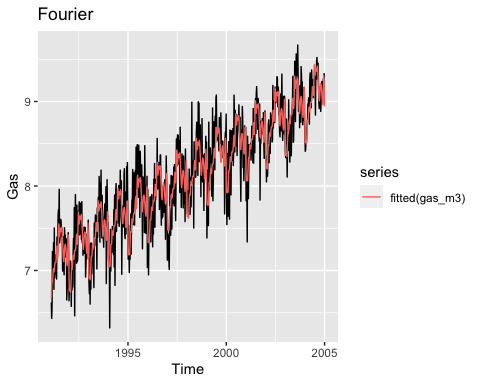
gas <- window(gasoline, end = 2005)  
  
  
gas\_m1 = tslm(gas ~ trend + fourier(gas, K=5))  
gas\_m2 = tslm(gas ~ trend + fourier(gas, K=10))  
gas\_m3 = tslm(gas ~ trend + fourier(gas, K=15))  
gas\_m4 = tslm(gas ~ trend + fourier(gas, K=20))  
  
autoplot(gas, ylab = "Gas",main= "Fourier") +autolayer(fitted(gas\_m1))



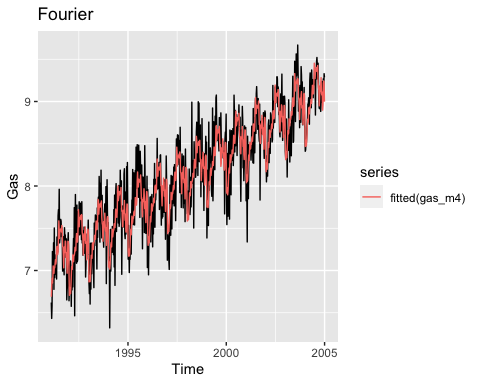
autoplot(gas, ylab = "Gas",main= "Fourier") +autolayer(fitted(gas\_m2))



autoplot(gas, ylab = "Gas",main= "Fourier") +autolayer(fitted(gas\_m3))



autoplot(gas, ylab = "Gas",main= "Fourier") +autolayer(fitted(gas\_m4))

 Чем больше пар Фурье используется, тем больше модель дает данные, похожие на исходные, но тренд не удается сохранить при этом.

*b. Select the appropriate number of Fourier terms to include by minimizing the AICc or CV value.*

CV(gas\_m1)

## CV AIC AICc BIC AdjR2   
## 7.553646e-02 -1.873234e+03 -1.872723e+03 -1.813596e+03 8.406928e-01

CV(gas\_m2)

## CV AIC AICc BIC AdjR2   
## 7.135754e-02 -1.915014e+03 -1.913441e+03 -1.809500e+03 8.516102e-01

CV(gas\_m3)

## CV AIC AICc BIC AdjR2   
## 7.190862e-02 -1.910231e+03 -1.906988e+03 -1.758841e+03 8.525942e-01

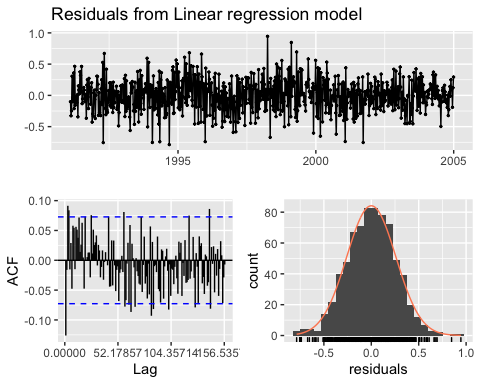
CV(gas\_m4)

## CV AIC AICc BIC AdjR2   
## 7.223834e-02 -1.908017e+03 -1.902469e+03 -1.710753e+03 8.540588e-01

gas\_m2 выбираем, так как у нее самые низкие CV и AIC.

*c. Check the residuals of the final model using the checkresiduals() function. Even though the residuals fail the correlation tests, the results are probably not severe enough to make much difference to the forecasts and prediction intervals. (Note that the correlations are relatively small, even though they are significant.)*

checkresiduals(gas\_m2)



##   
## Breusch-Godfrey test for serial correlation of order up to 104  
##   
## data: Residuals from Linear regression model  
## LM test = 155.45, df = 104, p-value = 0.0008135

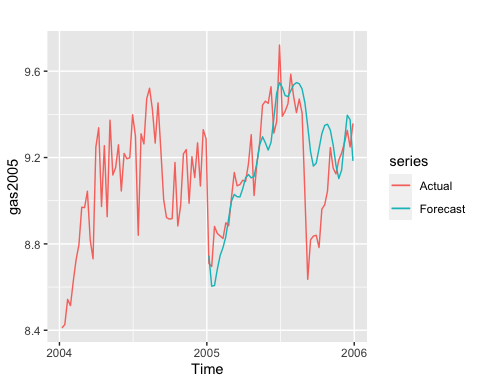
*d. To forecast using harmonic regression, you will need to generate the future values of the Fourier terms.*

gas\_fc <- forecast(gas\_m2, newdata=data.frame(fourier(gas,10,52)))  
gas\_fc

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 2005.014 8.745595 8.402358 9.088832 8.220249 9.270941  
## 2005.033 8.604075 8.260737 8.947413 8.078574 9.129575  
## 2005.052 8.607195 8.263839 8.950551 8.081666 9.132724  
## 2005.071 8.683215 8.339879 9.026550 8.157718 9.208712  
## 2005.090 8.746645 8.403404 9.089887 8.221293 9.271998  
## 2005.110 8.783003 8.439850 9.126156 8.257785 9.308221  
## 2005.129 8.833517 8.490370 9.176664 8.308309 9.358725  
## 2005.148 8.917498 8.574348 9.260647 8.392285 9.442710  
## 2005.167 8.997861 8.654729 9.340992 8.472676 9.523046  
## 2005.186 9.029128 8.686005 9.372252 8.503956 9.554301  
## 2005.205 9.018507 8.675381 9.361633 8.493330 9.543684  
## 2005.225 9.017572 8.674439 9.360705 8.492385 9.542760  
## 2005.244 9.056052 8.712918 9.399187 8.530863 9.581242  
## 2005.263 9.105523 8.762395 9.448651 8.580344 9.630703  
## 2005.282 9.121048 8.777925 9.464172 8.595876 9.646221  
## 2005.301 9.105879 8.762754 9.449005 8.580703 9.631055  
## 2005.320 9.112513 8.769382 9.455644 8.587329 9.637697  
## 2005.340 9.175080 8.831948 9.518211 8.649895 9.700265  
## 2005.359 9.259000 8.915873 9.602127 8.733822 9.784178  
## 2005.378 9.296291 8.953167 9.639415 8.771118 9.821464  
## 2005.397 9.268208 8.925083 9.611334 8.743032 9.793385  
## 2005.416 9.234527 8.891397 9.577657 8.709345 9.759710  
## 2005.435 9.270246 8.927115 9.613376 8.745062 9.795429  
## 2005.455 9.381554 9.038427 9.724681 8.856376 9.906731  
## 2005.474 9.497887 9.154762 9.841011 8.972713 10.023060  
## 2005.493 9.546877 9.203751 9.890002 9.021700 10.072053  
## 2005.512 9.524722 9.181592 9.867851 8.999539 10.049904  
## 2005.531 9.487091 9.143960 9.830221 8.961908 10.012274  
## 2005.550 9.482387 9.139260 9.825513 8.957209 10.007564  
## 2005.570 9.509124 9.165999 9.852248 8.983950 10.034297  
## 2005.589 9.536544 9.193418 9.879670 9.011367 10.061720  
## 2005.608 9.546848 9.203718 9.889978 9.021665 10.072030  
## 2005.627 9.541950 9.198820 9.885081 9.016767 10.067133  
## 2005.646 9.517407 9.174280 9.860533 8.992229 10.042584  
## 2005.665 9.453438 9.110314 9.796562 8.928265 9.978612  
## 2005.685 9.344405 9.001279 9.687531 8.819228 9.869581  
## 2005.704 9.226892 8.883762 9.570023 8.701709 9.752076  
## 2005.723 9.160315 8.817184 9.503446 8.635131 9.685499  
## 2005.742 9.174078 8.830951 9.517205 8.648901 9.699255  
## 2005.761 9.241612 8.898488 9.584736 8.716439 9.766785  
## 2005.780 9.310432 8.967306 9.653559 8.785255 9.835609  
## 2005.800 9.348790 9.005658 9.691922 8.823604 9.873976  
## 2005.819 9.354217 9.011085 9.697350 8.829031 9.879404  
## 2005.838 9.326591 8.983464 9.669718 8.801413 9.851768  
## 2005.857 9.258087 8.914964 9.601211 8.732915 9.783260  
## 2005.876 9.163285 8.820158 9.506412 8.638107 9.688463  
## 2005.895 9.102531 8.759394 9.445669 8.577338 9.627725  
## 2005.915 9.142445 8.799307 9.485583 8.617250 9.667640  
## 2005.934 9.276104 8.932975 9.619234 8.750923 9.801286  
## 2005.953 9.396214 9.053089 9.739340 8.871039 9.921390  
## 2005.972 9.374939 9.031803 9.718075 8.849747 9.900131  
## 2005.991 9.184286 8.841055 9.527518 8.658948 9.709624

*e. Plot the forecasts along with the actual data for 2005. What do you find?*

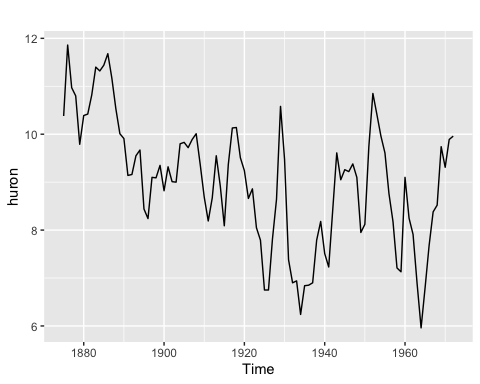
gas2005 = window(gasoline, start=2004, end=2006)  
autoplot(gas2005, series = "Actual")+ autolayer(gas\_fc$mean, series = "Forecast")

 Почти все данные находились в интервале 80% прогноза. Модель не могла предсказать внезапное падение осенью, оно было больше ожидаемого.

**7. Data set huron gives the water level of Lake Huron in feet from 1875-1972.**

*a. Plot the data and comment on its features.*

autoplot(huron)



head(huron)

## Time Series:  
## Start = 1875   
## End = 1880   
## Frequency = 1   
## [1] 10.38 11.86 10.97 10.80 9.79 10.39

Много колебаний, которые показывают цикл. И примерно до 1930 года был тренд на убывание, но после исчез.

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

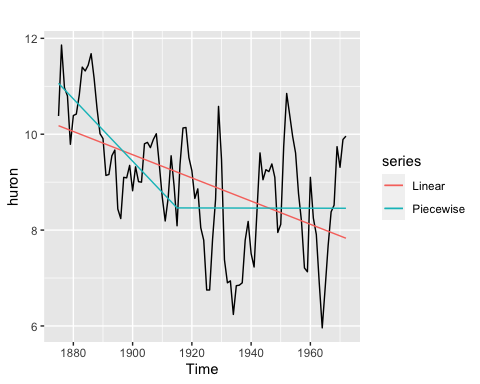
h\_lm <- tslm(huron ~ trend)  
  
t <- time(huron)  
t.break <- 1915  
t\_piece <- ts(pmax(0,t-t.break), start=1875)  
  
h\_pw <- tslm(huron ~ t + t\_piece)  
  
summary(h\_lm)

##   
## Call:  
## tslm(formula = huron ~ trend)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.50997 -0.72726 0.00083 0.74402 2.53565   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 10.202037 0.230111 44.335 < 2e-16 \*\*\*  
## trend -0.024201 0.004036 -5.996 3.55e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.13 on 96 degrees of freedom  
## Multiple R-squared: 0.2725, Adjusted R-squared: 0.2649   
## F-statistic: 35.95 on 1 and 96 DF, p-value: 3.545e-08

summary(h\_pw)

##   
## Call:  
## tslm(formula = huron ~ t + t\_piece)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.49626 -0.66240 -0.07139 0.85163 2.39222   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 132.90870 19.97687 6.653 1.82e-09 \*\*\*  
## t -0.06498 0.01051 -6.181 1.58e-08 \*\*\*  
## t\_piece 0.06486 0.01563 4.150 7.26e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.045 on 95 degrees of freedom  
## Multiple R-squared: 0.3841, Adjusted R-squared: 0.3711   
## F-statistic: 29.62 on 2 and 95 DF, p-value: 1.004e-10

autoplot(huron) +  
autolayer(fitted(h\_lm), series = "Linear") +  
autolayer(fitted(h\_pw), series = "Piecewise")

 Линейная модель и не cмогла отразить изменение тренда с 1915 года, в отличие от кусочно-линейной модели.

*c. Generate forecasts from these two models for the period upto 1980 and comment on these.*

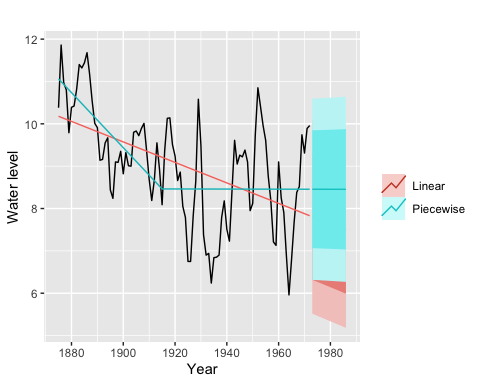
h = 14  
t\_new <- t[length(t)]+seq(h)  
t\_piece\_new <- t\_piece[length(t\_piece)]+seq(h)  
   
newdata <- cbind(t=t\_new,  
 t\_piece=t\_piece\_new) %>%  
 as.data.frame()  
  
h\_pw\_lm\_forecast <- forecast(  
 h\_pw,  
 newdata = newdata  
 )  
  
h\_lm\_forecast <- forecast(h\_lm, newdata = newdata)  
  
h\_lm\_forecast

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 1973 7.806127 6.317648 9.294605 5.516501 10.095752  
## 1974 7.781926 6.292536 9.271315 5.490899 10.072952  
## 1975 7.757724 6.267406 9.248043 5.465269 10.050179  
## 1976 7.733523 6.242259 9.224788 5.439613 10.027434  
## 1977 7.709322 6.217094 9.201550 5.413929 10.004715  
## 1978 7.685121 6.191912 9.178331 5.388219 9.982024  
## 1979 7.660920 6.166712 9.155128 5.362481 9.959359  
## 1980 7.636719 6.141494 9.131943 5.336717 9.936721  
## 1981 7.612518 6.116259 9.108776 5.310925 9.914110  
## 1982 7.588317 6.091007 9.085626 5.285107 9.891526  
## 1983 7.564116 6.065738 9.062494 5.259263 9.868968  
## 1984 7.539914 6.040451 9.039378 5.233391 9.846438  
## 1985 7.515713 6.015146 9.016280 5.207493 9.823933  
## 1986 7.491512 5.989825 8.993200 5.181569 9.801456

h\_pw\_lm\_forecast

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 1973 8.455119 7.063583 9.846655 6.314483 10.59575  
## 1974 8.454992 7.061518 9.848467 6.311374 10.59861  
## 1975 8.454866 7.059398 9.850333 6.308182 10.60155  
## 1976 8.454739 7.057225 9.852253 6.304906 10.60457  
## 1977 8.454612 7.054998 9.854227 6.301549 10.60768  
## 1978 8.454486 7.052717 9.856254 6.298109 10.61086  
## 1979 8.454359 7.050384 9.858334 6.294587 10.61413  
## 1980 8.454232 7.047997 9.860467 6.290984 10.61748  
## 1981 8.454106 7.045558 9.862653 6.287300 10.62091  
## 1982 8.453979 7.043067 9.864891 6.283535 10.62442  
## 1983 8.453852 7.040523 9.867181 6.279690 10.62801  
## 1984 8.453725 7.037927 9.869524 6.275766 10.63169  
## 1985 8.453599 7.035280 9.871917 6.271762 10.63544  
## 1986 8.453472 7.032582 9.874363 6.267679 10.63927

autoplot(huron) +  
autolayer(fitted(h\_lm), series = "Linear") +  
autolayer(fitted(h\_pw), series = "Piecewise") +  
autolayer(h\_lm\_forecast, series = "Linear") +  
autolayer(h\_pw\_lm\_forecast, series="Piecewise") +  
xlab("Year") + ylab("Water level") +  
guides(colour=guide\_legend(title=" "))

 Модель линейной регрессии показывает, что точечные прогнозы, а также верхняя и нижняя границы интервалов прогнозирования уменьшаются с течением времени. Она не отразила изменение тренда в 1915 году. Модель кусочно-линейной регрессии показывает, что точечные прогнозы и интервалы прогнозирования практически одинаковы во времени и отражает смену тренда 1915 года.

**8 задание** 