W271-2 – Spring 2016 – **Lab 3**

Juanjo Carin, Kevin Davis, Ashley Levato, Minghu Song

April 22, 2016

Contents

Instructions

- Thoroughly analyze the given dataset or data series. Detect any anomalies in each of the variables. Examine if any of the variables that may appear to be top- or bottom-coded.
- Your report needs to include a comprehensive graphical analysis
- Your analysis needs to be accompanied by detailed narrative. Just printing a bunch of graphs and econometric results will likely receive a very low score.
- Your analysis needs to show that your models are valid (in statistical sense).
- Your rationale of using certian metrics to choose models need to be provided. Explain the validity / pros / cons of the metric you use to choose your "best" model.
- Your rationale of any decisions made in your modeling needs to be explained and supported with empirical evidence.
- All the steps to arrive at your final model need to be shown and explained clearly.
- All of the assumptions of your final model need to be thoroughly tested and explained and shown to be valid. Don't just write something like, "the plot looks reasonable", or "the plot looks good", as different people interpret vague terms like "reasonable" or "good" differently.

Part 1

Modeling House Values

In Part 1, you will use the data set `houseValue.csv` to build a linear regression model, which includes the possible use of the instrumental variable approach, to answer a set of questions interested by a philanthropist group. You will also need to test hypotheses using these questions.

The philanthropist group hires a think tank to examine the relationship between the house values and neighborhood characteristics. For instance, they are interested in the extent to which houses in neighbhorhood with desirable features command higher values. They are specifically interested in environmental features, such as proximity to water body (i.e. lake, river, or ocean) or air quality of a region.

The think tank has collected information from tens of thousands of neighborhoods throughout the United States. They hire your group as contractors, and you are given a small sample and selected variables of the original data set collected to conduct an initial, proof-of-concept analysis. Many variables, in their original form or transfomed forms, that can explain the house values are included in the dataset. Analyze each of these variables as well as different combinations of them very carefully and use them (or a subset of them), in its original or transformed version, to build a linear regression model and test hypotheses to address the questions. Also address potential (statistical) issues that may be casued by omitted variables.

Based on the information in `homeValueData_VariableDescription.txt`, the variables and their meaning are:

- `crimeRate_pc`: crime rate per capital, measured by number of crimes per 1000 residents in neighborhood.
- `nonRetailBusiness`: the proportion of non-retail business acres per neighborhood.
- `withWater`: the neighborhood within 5 miles of a water body (lake, river, etc); 1 if true and 0 otherwise.
- `ageHouse`: proportion of house built before 1950.
- `distanceToCity`: distances to the nearest city (measured in miles).
- `pupilTeacherRatio`: average pupil-teacher ratio in all the schools in the neighborhood.
- `pctLowIncome`: percentage of low income household in the neighborhood
- `homeValue`: median price of single-family house in the neighborhood (measured in dollar).
- `pollutionIndex`: pollution index, scaled between 0 and 100, with 0 being the best and 100 being the worst (i.e. uninhabitable).
- `nBedRooms`: average number of bed rooms in the single family houses in the neighborhood.

First, we will load the data and conduct an exploratory analysis.

```
houseValue <- read.csv('houseValueData.csv', header = TRUE)
```

```
##             crimeRate_pc nonRetailBusiness withWater ageHouse
## nbr.val          400.000           400.000   400.000  400.000
## nbr.na             0.000             0.000     0.000    0.000
## skewness           4.962             0.288     3.435   -0.614
## kurtosis          33.982            -1.274     9.823   -0.947
## normtest.p         0.000             0.000     0.000    0.000
##             distanceToCity distanceToHighway pupilTeacherRatio pctLowIncome
## nbr.val            400.000           400.000           400.000      400.000
## nbr.na               0.000             0.000             0.000        0.000
## skewness             1.629             1.002            -0.772        0.967
## kurtosis             2.868            -0.871            -0.348        0.610
## normtest.p          0.000             0.000             0.000        0.000
##            homeValue pollutionIndex nBedRooms
## nbr.val      400.000        400.000   400.000
## nbr.na         0.000          0.000     0.000
## skewness       1.057          0.718     0.369
## kurtosis       1.545         -0.134     2.041
## normtest.p     0.000          0.000     0.000
```

The data consists of 400 observations (with no missing values) of 11 numeric variables: the ones mentioned above (median price of single-family houses in different neighborhoods and characteristics about those houses and neighborhoods) plus an additional one, not mentioned in the `txt` file:

- `distanceToHighway`: self-explanatory (and probably measured in miles, same as `distanceToCity`).

Based on the kurtosis, skewness (all of them far from zero to a greater or lesser extent) and the *p*-values of a normality test (all highly significant), none of the variables in the sample is normally distributed. That means they might benefit from transformation (potential transformations will be discussed as the exploratory analysis proceeds).

Table 1: Summary statistics of house values and features

Statistic	Mean	St. Dev.	Min	Pctl(25)	Median	Pctl(75)	Max
crimeRate_pc	3.763	8.872	0.006	0.083	0.266	3.675	88.976
nonRetailBusiness	0.112	0.070	0.007	0.051	0.097	0.181	0.277
withWater	0.068	0.251	0	0	0	0	1
ageHouse	68.932	27.977	2.900	45.675	77.950	94.150	100.000
distanceToCity	9.638	8.786	1.228	3.240	6.115	13.628	54.197
distanceToHighway	9.582	8.672	1	4	5	24	24
pupilTeacherRatio	21.391	2.168	15.600	19.900	21.900	23.200	25.000
pctLowIncome	15.795	9.341	2	8	14	21	49
homeValue	499,584.400	196,115.700	112,500	384,187.5	477,000	558,000	1,125,000
pollutionIndex	40.615	11.825	23.500	29.875	38.800	47.575	72.100
nBedRooms	4.266	0.719	1.561	3.883	4.193	4.582	6.780

Before plotting the distribution of each variable, we run a regression of the price value on all the other variables (after standardizing all to better compare their effects). This 1st regression model may not be the most appropriate one (data are not transformed, we won't check residulas, there may be multicollinearity...), but for the moment we just want to check if all the relationships make sense.

```r
houseValue.std <- houseValue %>% mutate_each(funs(scale))
model.1 <- lm(homeValue ~ ., houseValue.std)
names(sort(abs(model.1$coefficients), decreasing = T))
```

```
##  [1] "pctLowIncome"      "nBedRooms"         "pupilTeacherRatio"
##  [4] "pollutionIndex"    "distanceToCity"    "distanceToHighway"
##  [7] "crimeRate_pc"      "withWater"         "nonRetailBusiness"
## [10] "ageHouse"          "(Intercept)"
```

As shown in the table in the next page, the variables that have a stronger effect on the house value are `pctLowIncome` (a one standard deviation increase in it—which translates in a 9.3 point increase in the percentage of low income household in the neighborhood—decreases price by 0.37 standard deviation), then `nBedRooms` (a one standard deviation increase in it—which corresponds to 0.72 additional bedroomm, on average—increases price by 0.34 standard deviation), and so on. The variable that has a lower effect on the house value is the `ageHouse` (the roportion of houses built before 1950): though it may seem surprising that the effect is positive, it is not significant(ly different from zero), that could make sense: it's the age of each individual house, and not the average in the neighborhood, which should affect the price (and the age is not necessarily a bad feature: mansions of the 19th century are certainly more valuable than low-priced small houses, no matter how new they may be). The two variables that are significant only at the 10% level are `nonRetailBusiness` and `distanceToHighway`. `withWater` is significant at the 5% level, and `crimeRatio` at the 5% level; all these variables have the lowest effects (and the rest are significant at the 1% level. But what we matter most, at this early stage, it's the sign of the coefficients; and all of them make sense:

- A higher crime rate,
- more acres dedicated to non-retail businesses,
- farther distances to the nearest city,
- higher pupil-teacher ratios,
- a higher percentage of low-income households, and
- more pollution

all lead (other factors being equal) to lower house values. Similarly,

- closeness to a water body,
- a higher proportion of houses built before 1950 (already explained), and
- farther distance to the nearest highway

decrease the house value, on average.

As usual, all the standard errors are heteroskedasticity-robust.

Table 2: Regression summary (with standardized variables)

	Dependent variable:
	Median price ($) of single-family house
Crime rate per capita	−0.103**
	(0.032)
Proportion of non-retail business acres	−0.075·
	(0.045)
Water body less than 5 miles away	0.080*
	(0.040)
Proportion of houses built before 1950	0.026
	(0.059)
Distance (miles) to nearest city	−0.210***
	(0.045)
Distance (miles) to nearest highway	0.106·
	(0.057)
Average pupil-teacher ratio	−0.237***
	(0.032)
Percentage of low-income households	−0.369***
	(0.090)
Pollution index (0-100)	−0.220***
	(0.059)
Average number of bedrooms	0.345***
	(0.075)
Constant (intercept)	0.000
	(0.027)
F Statistic	74.444***
df	10; 389
Observations	400
R^2	0.724
Adjusted R^2	0.717
Residual Std. Error	0.532

·p<0.1; *p<0.05; **p<0.01; ***p<0.001

Next we plot the histogram (or bar chart) of the variables (and, in many cases, their log; using the log of a variable may make sense to narrow its range, satisfy the CLM assumptions more closely—e.g., reducing the skewness of the residuals—, model a non-linear—e.g., exponential—relationship, etc.). In principle, we don't care if the distribution of the regressors is normal; it's the distribution of the residuals which has to be (and that's not the strongest CLM assumption).

We begin with the dependent variable: `homeValue` is slightly right-skewed, with most values around the mean of approximately $500,000 and the right tail extending to the maximum of $1,125,000. A log transformation produces a distribution closer to normal, and we'll use it (besides, it makes a lot of sense for this regressand: the meaning of the coefficients—if not too high—will be a percentage change in the value).

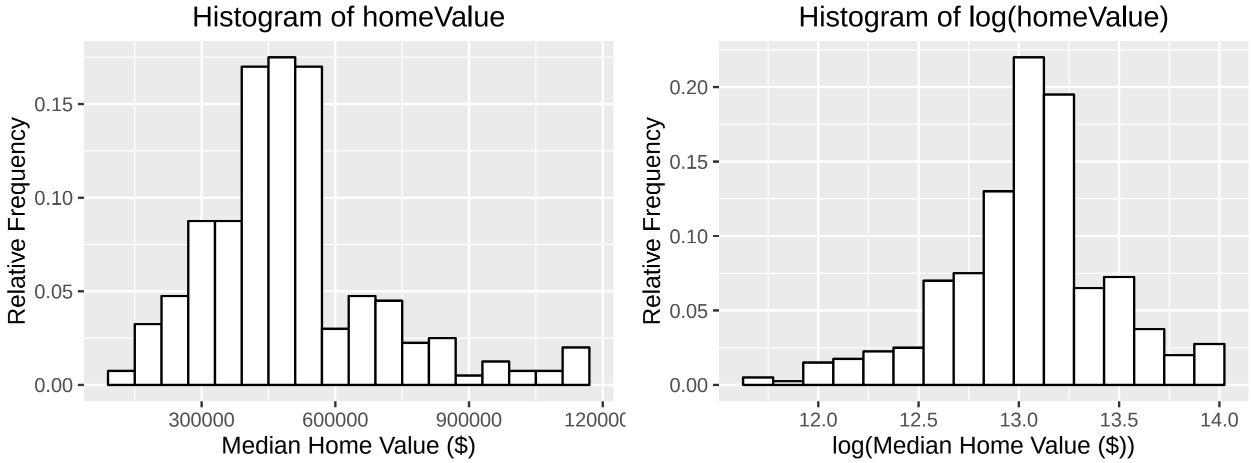


Figure 1: Histograms of home Value and its log

The crime rate variable is highly right-skewed, with most neighborhoods having a very low number of crimes per 1,000 residents, and a few having a high number. Using the log does not perfectly normalize that variable (the distribution is bimodal), but does a good enough job to use the log in this case.

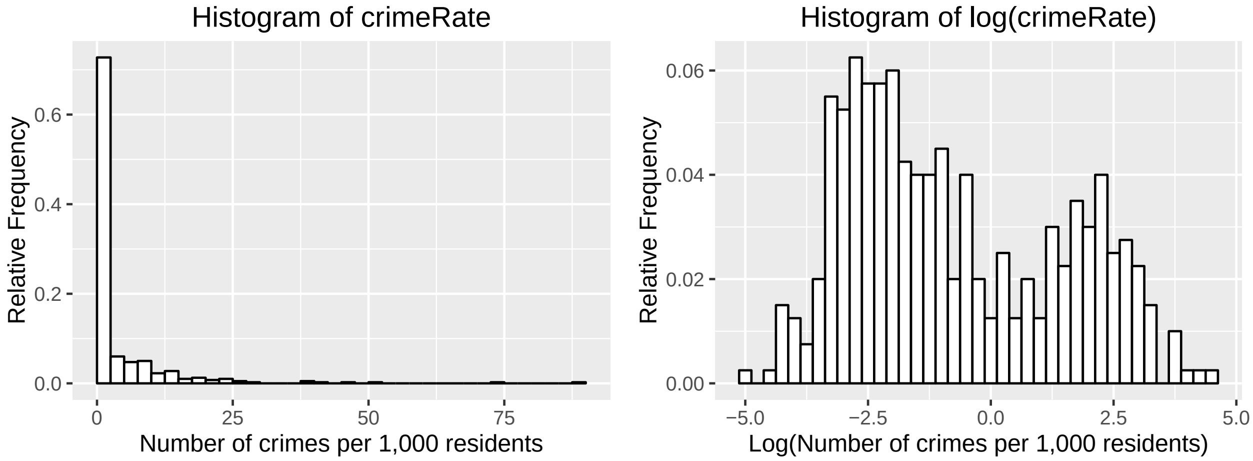


Figure 2: Histograms of Crime Rate and its log

As for the proportion of non-retail business acres per neighborhood, a high proportion of neighborhoods have non-retail business covering about 18% of their area, and most of the rest have much fewer non-retail businesses. A log transformation does not help to normalize this variable either.

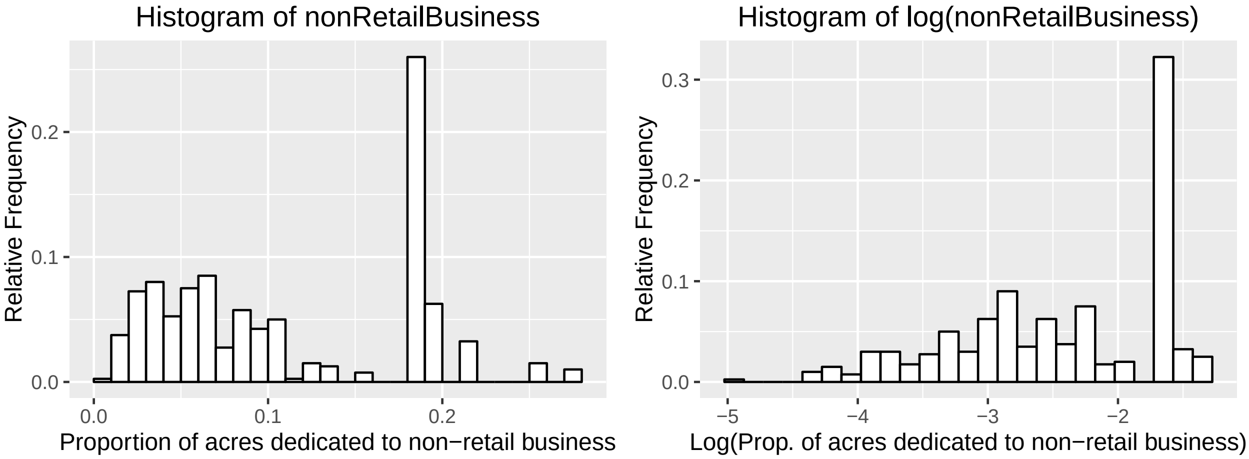


Figure 3: Histograms of non-retail Business acres and its log

Most neighborhoods are not located within 5 miles to a water body. Being near a lake or a river seems highly desirable, in principle, so it's a good candidate to have an effect on home values.

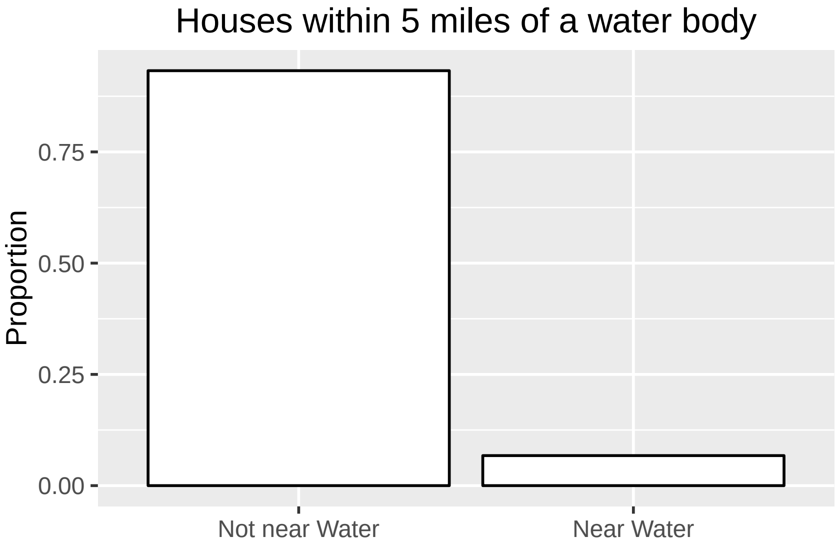


Figure 4: Char bart of proportion of houses within 5 miles of a water body

In almost 15% (13.75%) of the neighborhoods, more than 97.5% of the houses were built before 1950. If we lower that percentage of “hold houses” to 75%, that occurs in more than half of the neighborhoods (52.25%). In less than 10% of the neighborhoods (9.25%) only 25% of the houses or less are “old”“. Once again, a log transformation does not help to normalize the data.

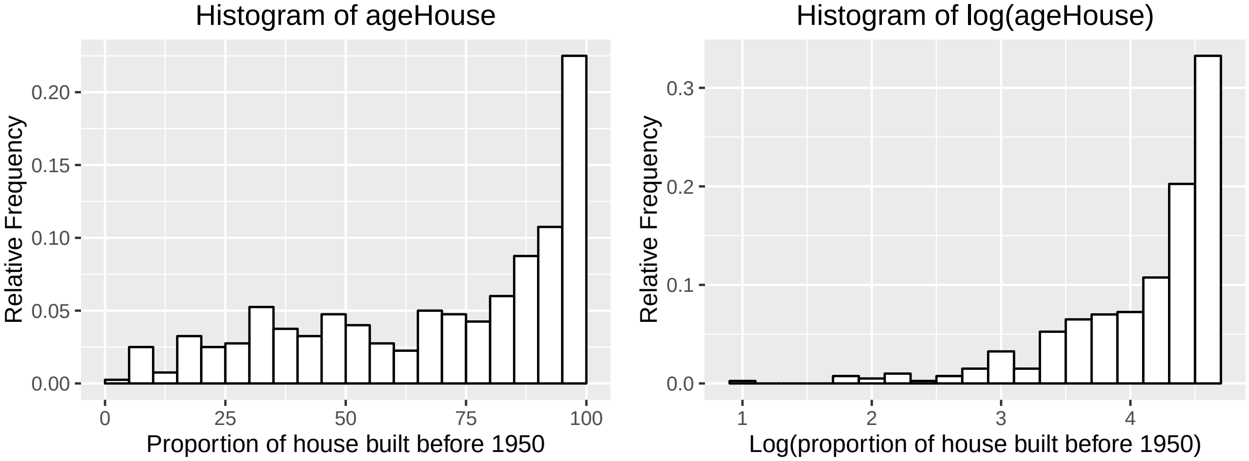


Figure 5: Histograms of Proportion of houses built before 1950 and its log

The distance from a neighborhood to nearby cities has a right-tailed distribution, with more than half of the neighborhoods (66.5%) within 10 miles of a city, and just 1% of them more than 40 miles away. Log transformation of this variable removed the skewness of the distribution and produced a more approximately normal distribution.

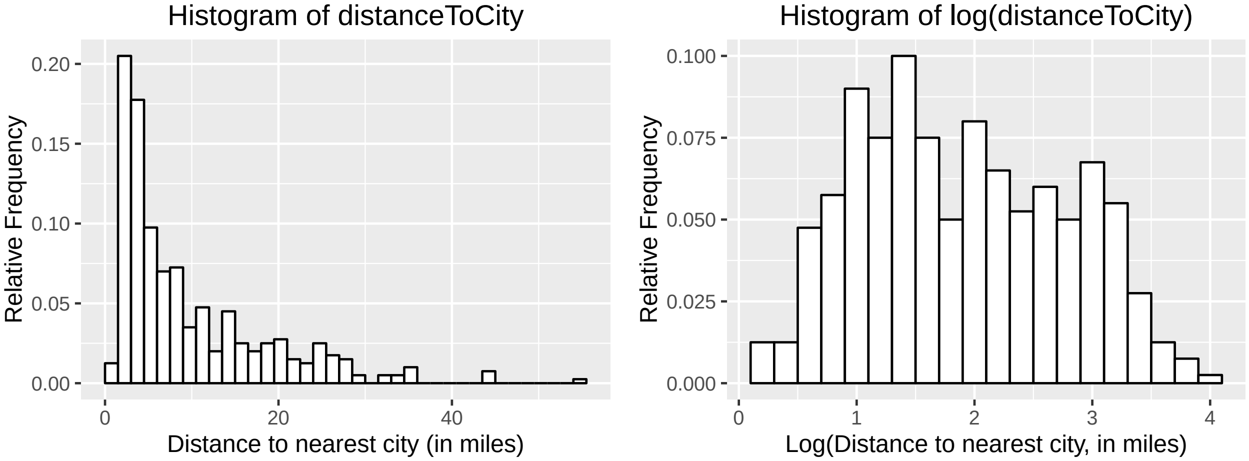


Figure 6: Histogram of distance to nearest city and its log

Since we’ll use the log for this variable, it seems appropriate to also use it for the next one (though the log does not normalize the data, as explained in the next page).

26% of the neighborhoods were exactly 24 miles from the nearest highway. There are only 9 unique values (the other possible distances go from 1 to 8 miles), which suggests us thinkg that this variable was probably rounded and factorized (losing part of its explanatory value). That makes the distribution to be strongly bimodal…even if it the log of the variable is used.

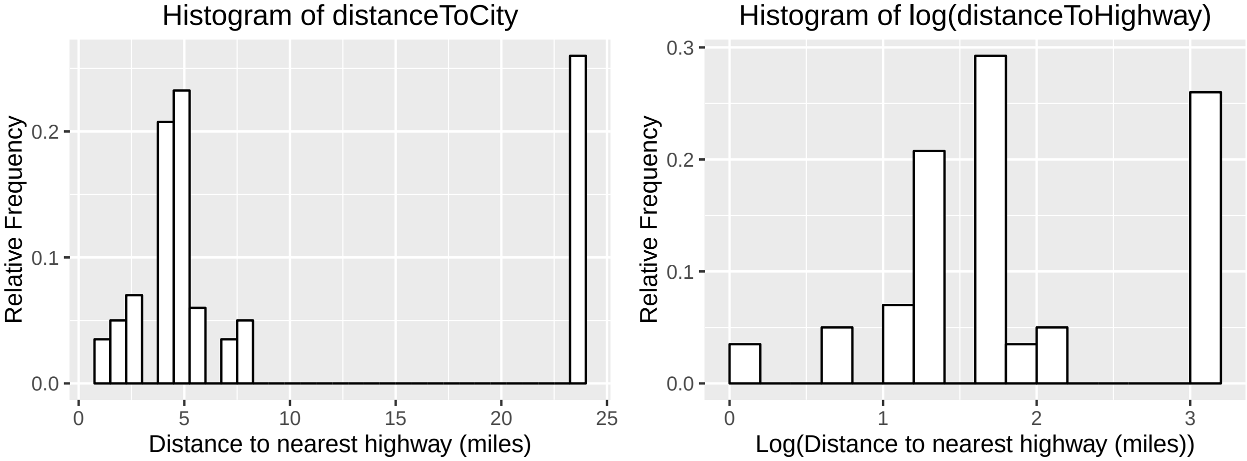


Figure 7: Histogram of distance to nearest highway and its log

The histogram of the average pupil-teacher ratio is left-skewed (and still is after using the log): many neighborhoods (27.5% of them has a ratio of 23.2 pupils per teacher; the other values (ranging from 15.6 to 25; and almost all lower) are approximately uniformly distributed). The distribution of the log of this variable looks pretty much the same.

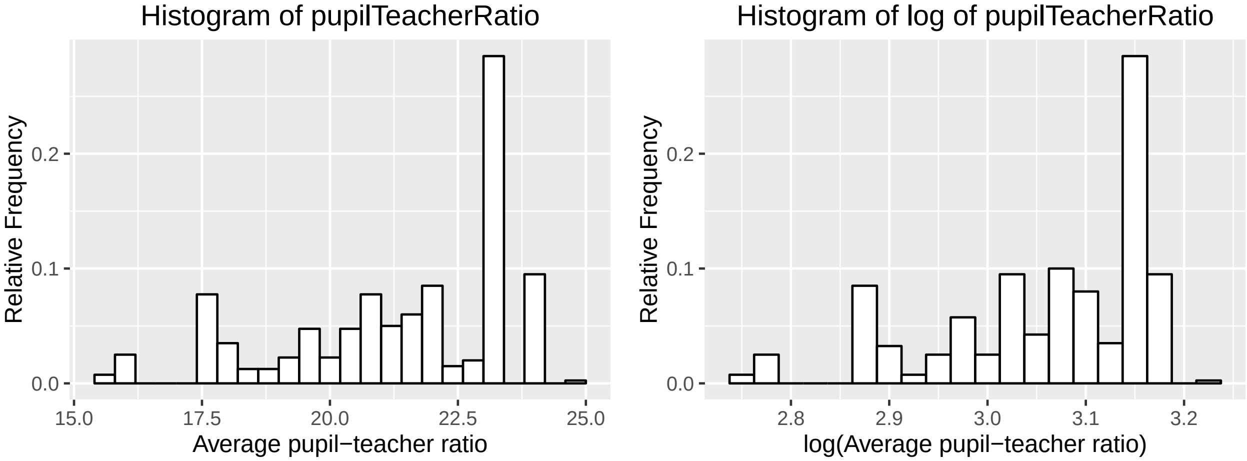


Figure 8: Histograms of Average pupil-teacher ratio and its log

The percentage of low-income households in a neighborhood displays a slightly right-skewed distribution. Since this is a percentage, keeping the data untransformed maintains the meaning of the regression coefficient: a unit increase means a 1% increase, which will result in a $100 \cdot beta_i$ increase (or decrease) in the home value (since we'll use the log of it).

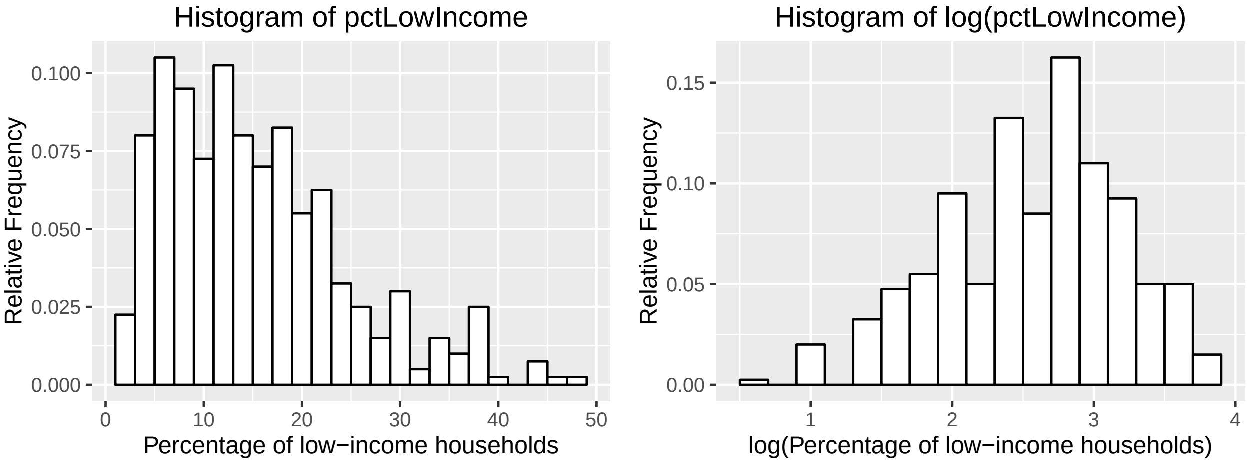


Figure 9: Histogram of percentage of low-income households and its log

The pollution index scores have a slightly-right tailed appearing distribution, with thin tails and evidence of multimodality. Log transformation of the pollution index reduced the right-skewness while still showing evidence of multimodality and thinner tails than a normal distribution.

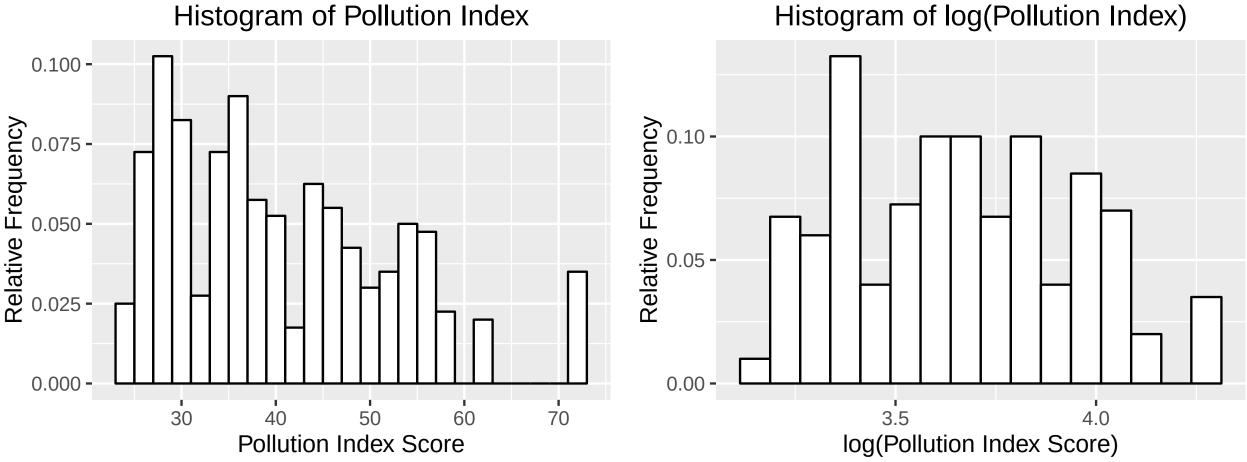


Figure 10: Histogram of percentage of low-income households and its log

The average number of bedrooms (with a mean of 4.3) is approximately normally distributed.

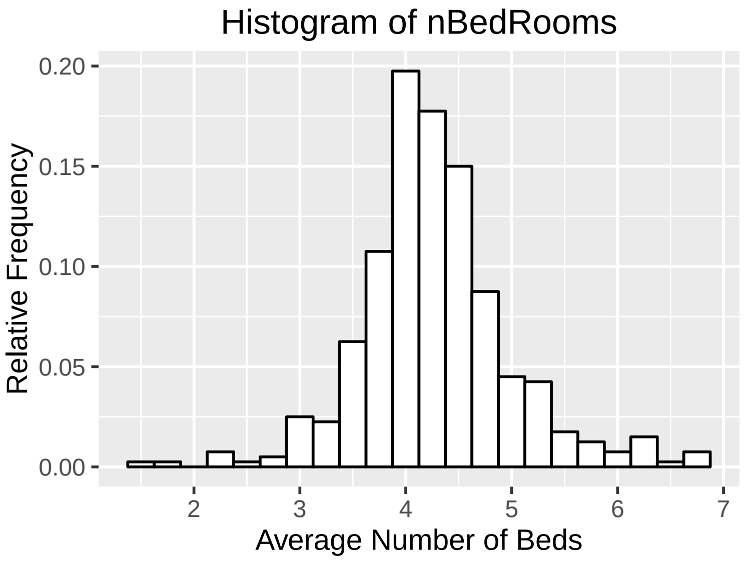


Figure 11: Histogram of average number of bedrooms

After visually inspecting each individual variable, we are also interested in how the other variables relate to the variable of interest, `homeValue`. First we apply the log to the (4) variables we previously mentioned (and change their names accordingly) and then we build a scatterplot matrix and run a simple regression of all the independent variables on `log_homeValue`.

```
vars_to_log <- c("homeValue", "crimeRate_pc", "distanceToCity",
                 "distanceToHighway")
houseValue.2 <- houseValue %>% mutate_each_(funs(log), vars_to_log) %>%
  setNames(c(paste0("log_", names(.)[1]), names(.)[2:4],
             paste0("log_", names(.)[5:6]), names(.)[7:8],
             paste0("log_", names(.)[9]), names(.)[10:11]))
```

```
regressors <- names(houseValue.2)[c(1:8, 10:11)]
model.list <- lapply(1:length(regressors), function(i)
  lm(as.formula(paste("log_homeValue ~", regressors[i])), houseValue.2))
```

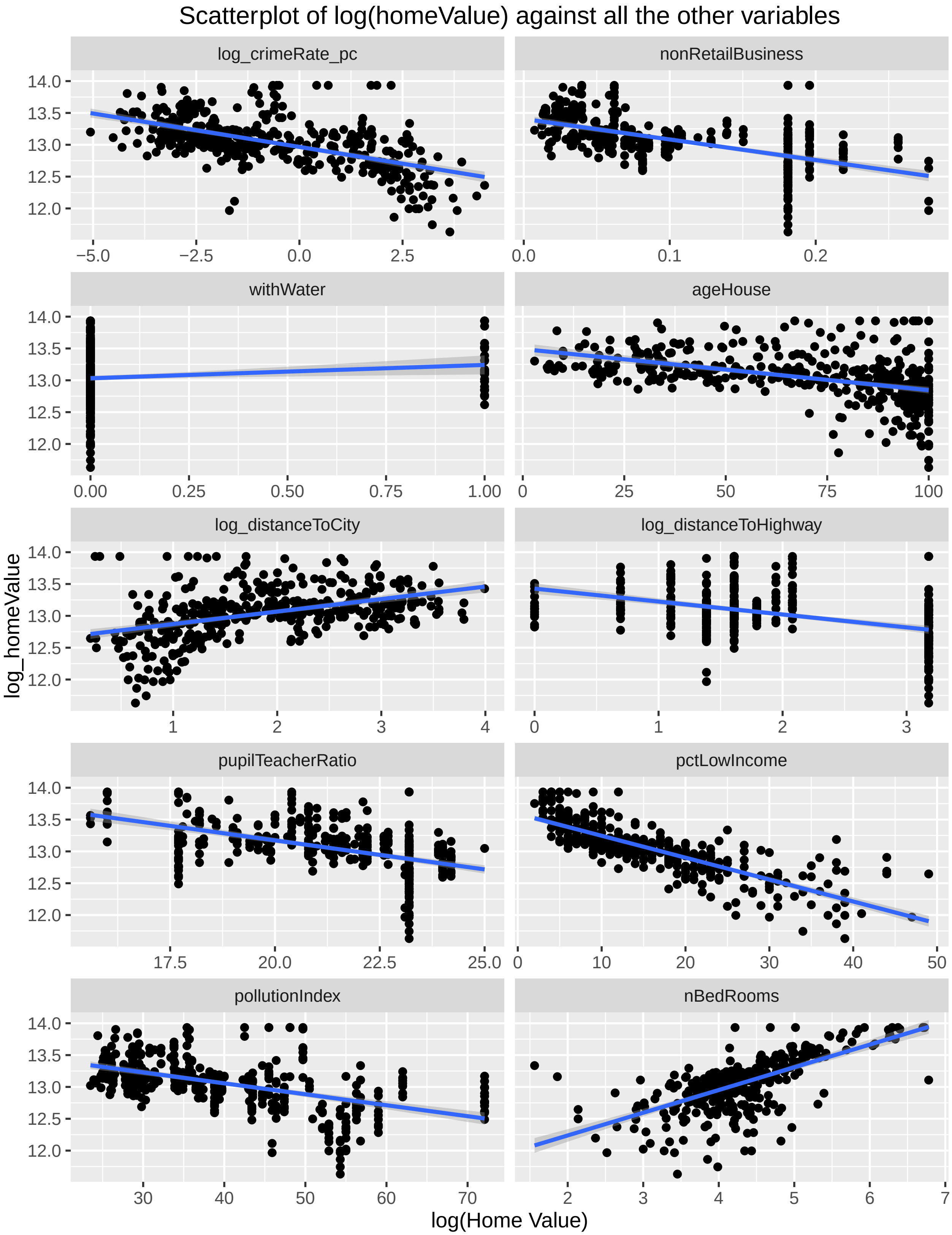


Figure 12: Scatter Plot of log(homeValue) against all the other variables

Table 3: Simple regression summary of log(homeValue)

	Dependent variable:									
	log(Median price ($) of single-family house)									
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
log(crime rate)	−0.105***									
	(0.008)									
Prop. NR business		−3.230***								
		(0.253)								
Water<5 miles			0.210**							
			(0.077)							
Prop. houses<1950				−0.006***						
				(0.001)						
log(dist. city)					0.196***					
					(0.023)					
log(dist. highway)						−0.203***				
						(0.023)				
Avg p-t ratio							−0.091***			
							(0.008)			
% low-inc. house								−0.034***		
								(0.002)		
Pollution									−0.017***	
									(0.002)	
no. bedrooms										0.357***
										(0.029)
Constant	12.966***	13.406***	13.032***	13.491***	12.675***	13.427***	14.998***	13.590***	13.745***	11.524***
	(0.019)	(0.026)	(0.020)	(0.035)	(0.054)	(0.042)	(0.174)	(0.029)	(0.057)	(0.129)
F Statistic	169.087***	162.378***	7.469**	121.735***	73.192***	79.045***	125.722***	306.988***	126.672***	148.118***
df	1; 398	1; 398	1; 398	1; 398	1; 398	1; 398	1; 398	1; 398	1; 398	1; 398
Observations	400	400	400	400	400	400	400	400	400	400
R^2	0.328	0.321	0.018	0.207	0.184	0.195	0.249	0.656	0.263	0.417
Adjusted R^2	0.326	0.319	0.015	0.205	0.182	0.193	0.247	0.655	0.261	0.416
Residual Std. Error	0.326	0.328	0.394	0.354	0.359	0.357	0.345	0.233	0.341	0.303

·p<0.1; *p<0.05; **p<0.01; ***p<0.001

The Table in the previous page shows that each variable, when not controlling for any other, is a good predictor of the log of the median house value (much better than just the mean of it). This fact may complicate our effort to select one of these variables as an instrument, as being unrelated to the outcome variable is one condition of the exclusion restriction for an IV approach. However, this does not necessarily preclude all the variables from being used as an instrument, as some variables may not be significant when controlling for other variables.

The only coefficients that change their sign when running a simple regression (as opposed to a multiple regression when all independent variables are used) are the distance to the nearest city and highway, respectively. As a result, if we don't control for other factors, a further distance to the nearest city increases the median value of a house, and the opposite occurs with the nearest highway. This is not due to the use of logarithms (the same happens if we don't apply them to either the home value or the distance) but because of the inclusion of other variables, some of them which may be related to those 2 variables.

Since we are particularly interested on the impact of environmental variables on the value of homes, we also want to understand how those variables relate to the other variables in the dataset. As shown in the following 4 pages, `pollutionIndex` is highly related (positively or negatively) with all the other independent variables, while `withWater` is only related with a few of them (`pollutionIndex` itself, `ageHouse`, and `nonRetailBusiness`, at the 5% level). The former fact provides evidence that estimating the effects of pollution on home values requires controlling for a number of potential confounding variables.

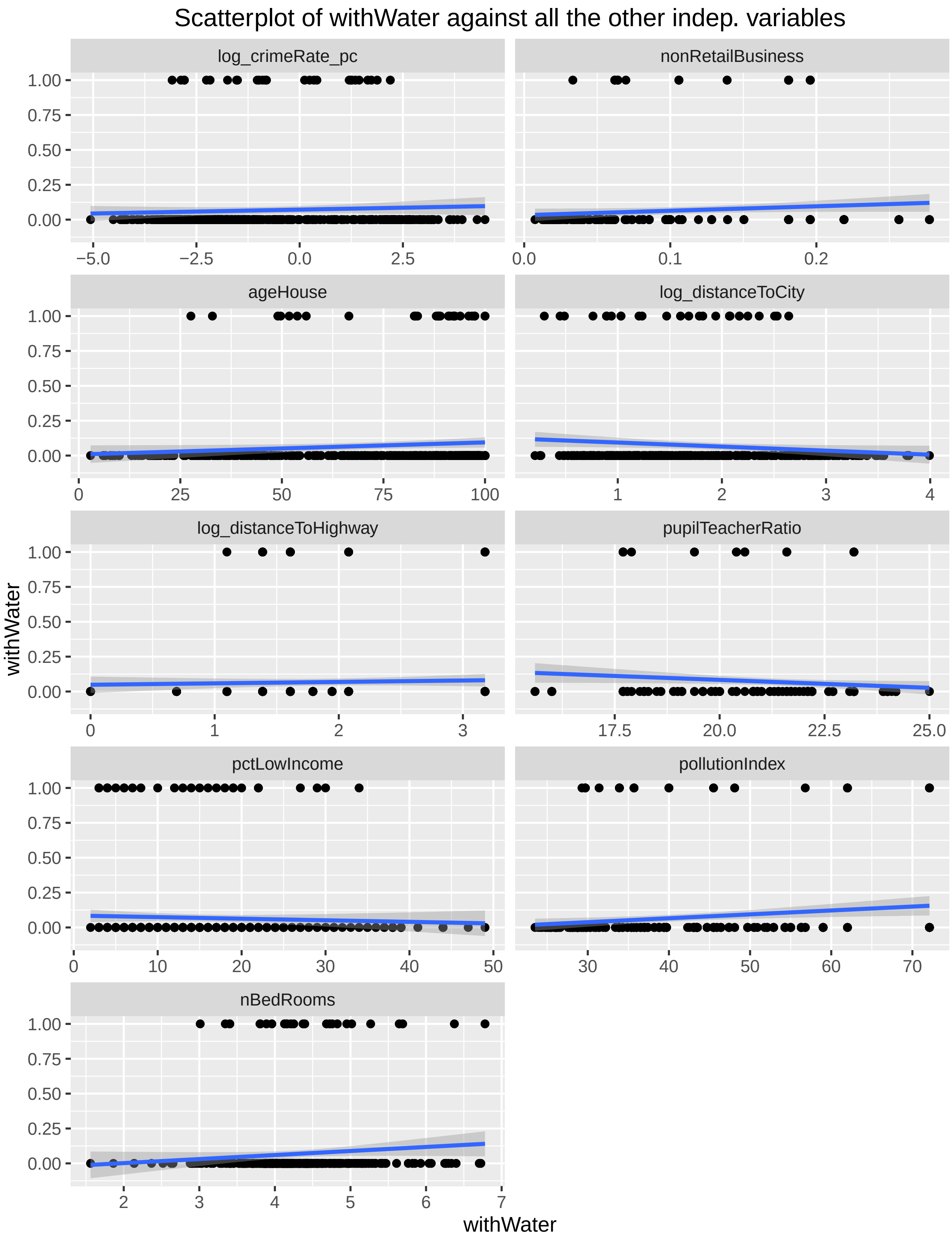


Figure 13: Scatter Plot of withWater against all the other independent variables

Table 4: Simple regression summary of withWater

	Dependent variable:								
	Proximity (< 5 miles) to a of a water body								
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
log(crime rate)	0.006								
	(0.005)								
Prop. NR business		0.318*							
		(0.161)							
Prop. houses<1950			0.001*						
			(0.0004)						
log(dist. city)				−0.029*					
				(0.013)					
log(dist. highway)					0.010				
					(0.013)				
Avg p-t ratio						−0.011$^{\cdot}$			
						(0.006)			
% low-inc. house							−0.001		
							(0.001)		
Pollution								0.003*	
								(0.001)	
no. bedrooms									0.029
									(0.022)
Constant	0.072***	0.032$^{\cdot}$	0.008	0.122***	0.048$^{\cdot}$	0.310*	0.085***	−0.046	−0.056
	(0.014)	(0.018)	(0.024)	(0.031)	(0.025)	(0.132)	(0.026)	(0.053)	(0.091)
F Statistic	1.383	3.895*	5.207*	5.358*	0.642	3.662$^{\cdot}$	0.756	4.094*	1.782
df	1; 398	1; 398	1; 398	1; 398	1; 398	1; 398	1; 398	1; 398	1; 398
Observations	400	400	400	400	400	400	400	400	400
R^2	0.002	0.008	0.009	0.010	0.001	0.010	0.002	0.017	0.007
Adjusted R^2	−0.0002	0.005	0.007	0.008	−0.001	0.007	−0.001	0.015	0.004
Residual Std. Error	0.251	0.251	0.250	0.250	0.251	0.250	0.251	0.249	0.251

$\cdot$p<0.1; *p<0.05; **p<0.01; ***p<0.001

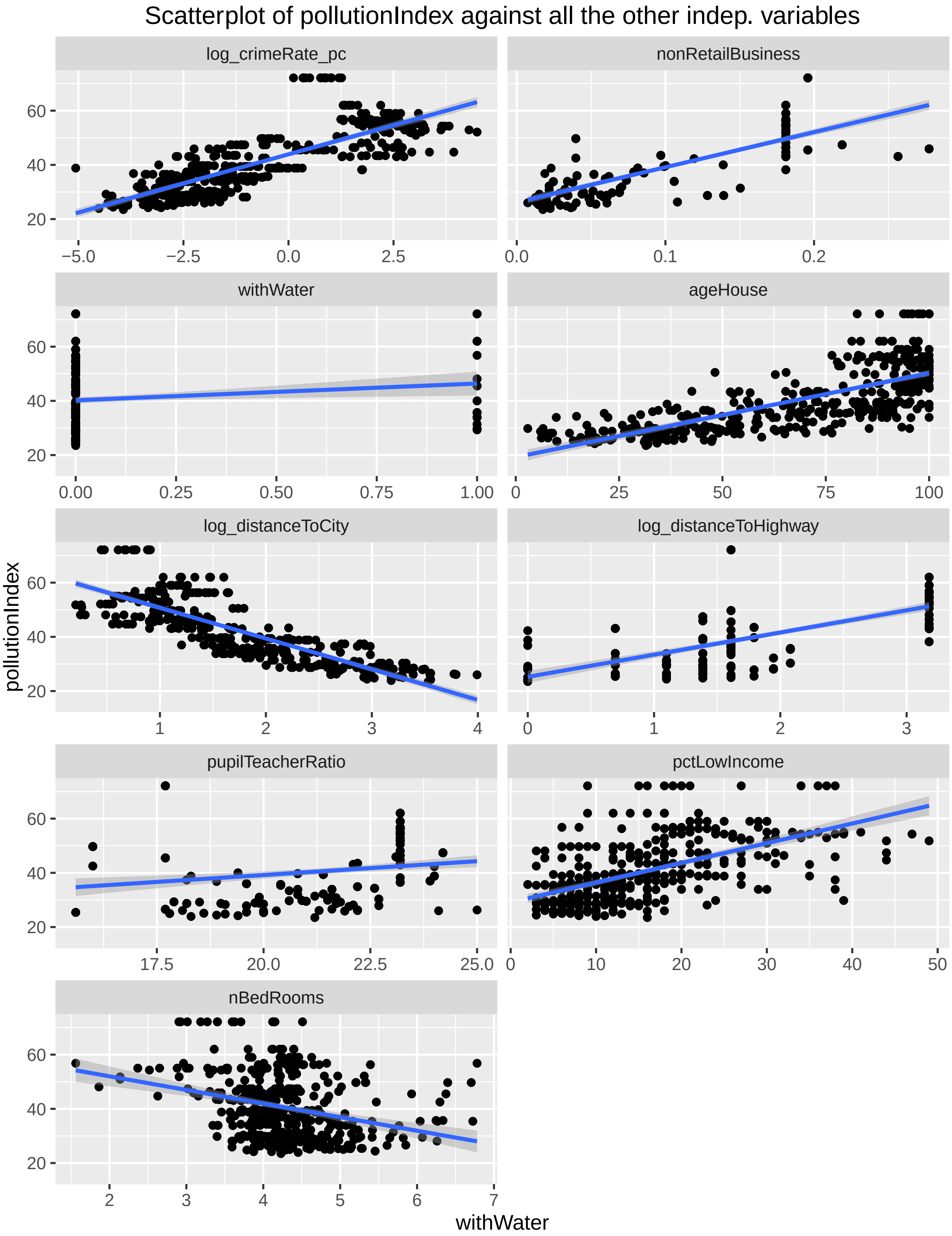


Figure 14: Scatter Plot of pollutionIndex against all the other independent variables

Table 5: Simple regression summary of pollutionIndex

	Dependent variable:								
	Pollution Index								
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
log(crime rate)	4.295***								
	(0.162)								
Prop. NR business		129.499***							
		(6.727)							
Water<5 miles			6.207*						
			(3.071)						
Prop. houses<1950				0.310***					
				(0.013)					
log(dist. city)					−11.335***				
					(0.394)				
log(dist. highway)						8.183***			
						(0.393)			
Avg p-t ratio							1.020**		
							(0.346)		
% low-inc. house								0.726***	
								(0.055)	
no. bedrooms									−5.023***
									(0.830)
Constant	43.890***	26.175***	40.196***	19.243***	62.062***	25.238***	18.786*	29.141***	62.042***
	(0.437)	(0.645)	(0.593)	(0.757)	(0.967)	(0.970)	(7.650)	(0.901)	(3.594)
F Statistic	701.991***	370.554***	4.087*	576.467***	829.270***	433.845***	8.710**	172.149***	36.639***
df	1; 398	1; 398	1; 398	1; 398	1; 398	1; 398	1; 398	1; 398	1; 398
Observations	400	400	400	400	400	400	400	400	400
R^2	0.618	0.581	0.017	0.538	0.692	0.358	0.035	0.329	0.093
Adjusted R^2	0.617	0.580	0.015	0.537	0.692	0.356	0.033	0.328	0.091
Residual Std. Error	7.320	7.667	11.737	8.047	6.568	9.489	11.631	9.697	11.275

·p<0.1; *p<0.05; **p<0.01; ***p<0.001

To select a candidate for a good model we'll make use of a stepwise selection (using the BIC rather than the AIC—though the output of R's `step` does not change the name of the criterion—to put a greater penalty in the number of regressors).

```
full.model <- lm(log_homeValue ~ ., data = houseValue.2)
null.model <- lm(log_homeValue ~ 1, data = houseValue.2)
# k=log(n) (instead of default k=2) to use  BIC rather than AIC
stepwise.reg <- step(null.model, scope = list(lower = null.model,
                                              upper=full.model),
                     direction="both", k = log(400))
```

```
## Start:  AIC=-734.15
## log_homeValue ~ 1
##
##                        Df Sum of Sq    RSS      AIC
## + pctLowIncome          1    41.239 21.634 -1154.90
## + nBedRooms             1    26.243 36.629  -944.26
## + log_crimeRate_pc      1    20.617 42.255  -887.11
## + nonRetailBusiness     1    20.158 42.714  -882.79
## + pollutionIndex        1    16.514 46.359  -850.04
## + pupilTeacherRatio     1    15.624 47.249  -842.43
## + ageHouse              1    13.002 49.870  -820.83
## + log_distanceToHighway 1    12.270 50.602  -815.01
## + log_distanceToCity    1    11.586 51.287  -809.63
## + withWater             1     1.106 61.767  -735.26
## <none>                              62.873  -734.15
##
## Step:  AIC=-1154.9
## log_homeValue ~ pctLowIncome
##
##                        Df Sum of Sq    RSS      AIC
## + pupilTeacherRatio     1     2.202 19.431 -1191.86
## + nBedRooms             1     1.958 19.675 -1186.87
## + withWater             1     0.611 21.023 -1160.37
## + nonRetailBusiness     1     0.490 21.144 -1158.08
## + log_crimeRate_pc      1     0.425 21.208 -1156.85
## + log_distanceToHighway 1     0.388 21.245 -1156.15
## <none>                              21.634 -1154.90
## + pollutionIndex        1     0.214 21.420 -1152.89
## + log_distanceToCity    1     0.154 21.480 -1151.77
## + ageHouse              1     0.075 21.559 -1150.30
## - pctLowIncome          1    41.239 62.873  -734.15
##
## Step:  AIC=-1191.86
## log_homeValue ~ pctLowIncome + pupilTeacherRatio
##
##                        Df Sum of Sq    RSS      AIC
## + nBedRooms             1    1.5804 17.851 -1219.80
## + withWater             1    0.4262 19.005 -1194.74
## + pollutionIndex        1    0.3051 19.126 -1192.20
## <none>                              19.431 -1191.86
## + log_distanceToCity    1    0.1714 19.260 -1189.41
## + nonRetailBusiness     1    0.1576 19.274 -1189.12
## + ageHouse              1    0.1322 19.299 -1188.60
## + log_crimeRate_pc       1    0.1187 19.313 -1188.32
## + log_distanceToHighway  1    0.0249 19.406 -1186.38
## - pupilTeacherRatio      1    2.2023 21.634 -1154.90
## - pctLowIncome           1   27.8174 47.249  -842.43
##
## Step:  AIC=-1219.8
## log_homeValue ~ pctLowIncome + pupilTeacherRatio + nBedRooms
##
##                         Df Sum of Sq    RSS     AIC
## + pollutionIndex         1    0.4309 17.420 -1223.6
## + log_crimeRate_pc       1    0.3607 17.490 -1222.0
## + withWater              1    0.3287 17.522 -1221.2
## <none>                               17.851 -1219.8
## + nonRetailBusiness      1    0.1658 17.685 -1217.5
## + log_distanceToHighway  1    0.1655 17.685 -1217.5
## + log_distanceToCity     1    0.0490 17.802 -1214.9
## + ageHouse               1    0.0047 17.846 -1213.9
## - nBedRooms              1    1.5804 19.431 -1191.9
## - pupilTeacherRatio      1    1.8245 19.675 -1186.9
## - pctLowIncome           1   12.9288 30.780 -1007.9
##
## Step:  AIC=-1223.58
## log_homeValue ~ pctLowIncome + pupilTeacherRatio + nBedRooms +
##     pollutionIndex
##
##                         Df Sum of Sq    RSS     AIC
## + log_distanceToCity     1    1.1615 16.259 -1245.2
## + withWater              1    0.4967 16.923 -1229.2
## + ageHouse               1    0.3357 17.084 -1225.4
## <none>                               17.420 -1223.6
## - pollutionIndex         1    0.4309 17.851 -1219.8
## + log_crimeRate_pc       1    0.0397 17.380 -1218.5
## + log_distanceToHighway  1    0.0080 17.412 -1217.8
## + nonRetailBusiness      1    0.0015 17.419 -1217.6
## - nBedRooms              1    1.7062 19.126 -1192.2
## - pupilTeacherRatio      1    1.9121 19.332 -1187.9
## - pctLowIncome           1    7.5976 25.018 -1084.8
##
## Step:  AIC=-1245.19
## log_homeValue ~ pctLowIncome + pupilTeacherRatio + nBedRooms +
##     pollutionIndex + log_distanceToCity
##
##                         Df Sum of Sq    RSS     AIC
## + withWater              1    0.4720 15.787 -1251.0
## <none>                               16.259 -1245.2
## + log_crimeRate_pc       1    0.1559 16.103 -1243.0
## + nonRetailBusiness      1    0.0966 16.162 -1241.6
## + log_distanceToHighway  1    0.0199 16.239 -1239.7
## + ageHouse               1    0.0152 16.243 -1239.6
## - log_distanceToCity     1    1.1615 17.420 -1223.6
## - nBedRooms              1    1.2952 17.554 -1220.5
## - pollutionIndex         1    1.5433 17.802 -1214.9
## - pupilTeacherRatio      1    2.2115 18.470 -1200.2
## - pctLowIncome           1    8.5804 24.839 -1081.7
```

```
##
## Step:  AIC=-1250.98
## log_homeValue ~ pctLowIncome + pupilTeacherRatio + nBedRooms +
##     pollutionIndex + log_distanceToCity + withWater
##
##                         Df Sum of Sq    RSS     AIC
## <none>                               15.787 -1251.0
## + log_crimeRate_pc       1    0.1394 15.647 -1248.5
## + nonRetailBusiness      1    0.1245 15.662 -1248.2
## + log_distanceToHighway  1    0.0181 15.768 -1245.5
## + ageHouse               1    0.0086 15.778 -1245.2
## - withWater              1    0.4720 16.259 -1245.2
## - log_distanceToCity     1    1.1368 16.923 -1229.2
## - nBedRooms              1    1.2195 17.006 -1227.2
## - pollutionIndex         1    1.7165 17.503 -1215.7
## - pupilTeacherRatio      1    2.0491 17.836 -1208.2
## - pctLowIncome           1    8.2783 24.065 -1088.3
```

Table 6: Regression model of log(homeValue)

	Dependent variable:
	log(Median price ($) of single-family house)
Percentage of low-income households	-0.025^{***}
	(0.003)
Average pupil-teacher ratio	-0.037^{***}
	(0.005)
Average number of bedrooms	0.101^{***}
	(0.027)
Pollution index (0-100)	-0.010^{***}
	(0.002)
log(Distance (miles) to nearest city)	-0.115^{***}
	(0.026)
Water body less than 5 miles away	0.140^{***}
	(0.040)
Constant (intercept)	14.419^{***}
	(0.224)
F Statistic	140.988^{***}
df	6; 393
Observations	400
R^2	0.749
Adjusted R^2	0.745
Residual Std. Error	0.200

·p<0.1; *p<0.05; **p<0.01; ***p<0.001

Thus, the model is:

$$\begin{aligned}\log(homeValue) =& \beta_0 + \beta_1 \cdot pctLowIncome + \beta_2 \cdot pupilTeacherRatio + \beta_3 \cdot nBedRooms \\ &+ \beta_4 \cdot pollutionIndex + \beta_5 \cdot \log(distanceToCity) + \beta_6 \cdot withWater + \varepsilon\end{aligned}$$

And (because we've used the log of the dependent variable and one of the independent variables), the coefficients tell us that:

- when the percentage of low-income households in the neighborhood increases by 1, the median value of a house in that neighborhood decreases 2.5%.

- when the number of pupils per teahcer increases by 1, the median home value decreases by 3.7%.
- an additional bedroom increases the value of the house by approximately 10.1%.
- when the pollution index increases by 1, the median home value decreases by 1.0% (since the standard deviation of the Pollution Index is 11.8, an increase of 1 standard deviation in that index would decrease the value of the house by approximately 12.2%).
- when the distance to the nearest city increases by 1% (i.e., almost 0.1 mile, since the mean of that variable is 9.6 miles), the median home value decreases by 11.5%.
- the proximity to a water body increases the home value by 14.0%.

I.e., not only the variables are statistically significant but also have **practical significance**.

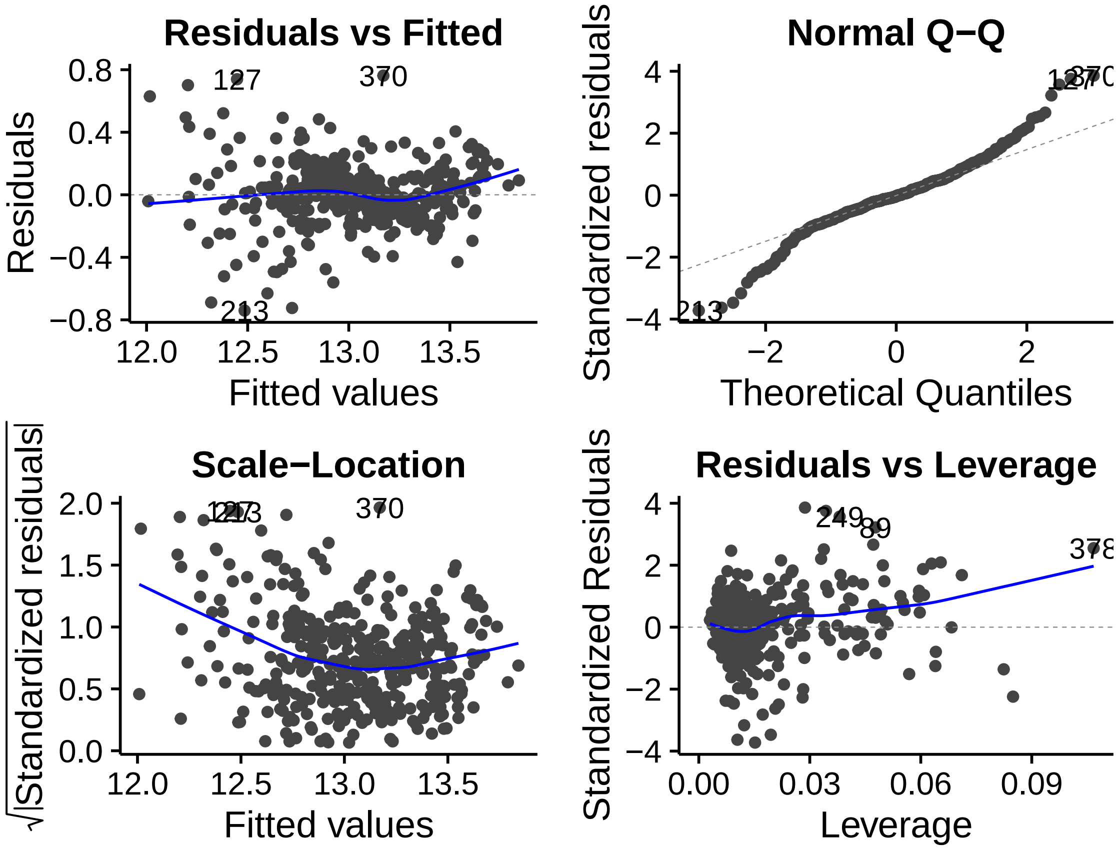


Figure 15: Residual diagnostics

The previous *F* statistic tell us the overall significance of the regression model we've built. We can also test whether each individual regressor helps to explain the variability in home values (controlling for other factors; in other words, if it's worth keeping it in the model). As expected, all the included regressors are significantly different from zero.

```
hyp <- lapply(names(stepwise.reg$coefficients)[-1], function(x)
  linearHypothesis(stepwise.reg, x, vcov = vcovHC))
null.hyp_p <- unlist(lapply(c(1:(length(stepwise.reg$coefficients) - 1)),
                            function(i) (hyp[[i]])$`Pr(>F)`[2]))
names(null.hyp_p) <- names(stepwise.reg$coefficients)[-1]
null.hyp_p
```

```
##       pctLowIncome  pupilTeacherRatio          nBedRooms
##       2.302064e-15       6.550973e-12       2.147518e-04
##     pollutionIndex log_distanceToCity          withWater
##       1.009110e-07       1.635181e-05       5.168360e-04
```

We are worried that `pollutionIndex` may be correlated with many omitted variables (and consequently with the residuals of our current model), and hence it could bias the other coefficients. We try 2 remaining variables (`log_distanceToHighway` and `nonRetailBusiness`) that should be highly correlated with the pollution, but not with other factors related to it (that remain in the residuals).

(The relationship of `distanceToHighway` with the pollution seems obvious; as for `nonRetailBusiness`, more acres dedicated to this kind of businesses may increase the likelihood of some of them being the type that has a huge impact on environment, such as heavy manufacturing, open-pit mining, etc.)

If we add these 2 variables to the model, they don't add explanatory power (i.e., they don't help predict some variance left in the residuals). That makes us confident that both may be unrelated to the residuals (though we cannot prove that):

```
new_model <- update(stepwise.reg, . ~ . + log_distanceToHighway +
                      nonRetailBusiness)
linearHypothesis(new_model, "log_distanceToHighway")
```

```
## Linear hypothesis test
##
## Hypothesis:
## log_distanceToHighway = 0
##
## Model 1: restricted model
## Model 2: log_homeValue ~ pctLowIncome + pupilTeacherRatio + nBedRooms +
##     pollutionIndex + log_distanceToCity + withWater + log_distanceToHighway +
##     nonRetailBusiness
##
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    392 15.662
## 2    391 15.646  1  0.015736 0.3932  0.531
```

```
linearHypothesis(new_model, "nonRetailBusiness")
```

```
## Linear hypothesis test
##
## Hypothesis:
## nonRetailBusiness = 0
##
## Model 1: restricted model
## Model 2: log_homeValue ~ pctLowIncome + pupilTeacherRatio + nBedRooms +
##     pollutionIndex + log_distanceToCity + withWater + log_distanceToHighway +
##     nonRetailBusiness
##
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1    392 15.768
## 2    391 15.646  1   0.12212 3.0518 0.08143 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

On the other hand, they are highly correlated to `pollutionIndex`:

Table 7: Regression model of log(homeValue) adding possibleIVs

	Dependent variable:
	log(Median price ($) of single-family house)
Percentage of low-income households	-0.024^{***}
	(0.003)
Average pupil-teacher ratio	-0.032^{***}
	(0.005)
Average number of bedrooms	0.099^{***}
	(0.028)
Pollution index (0-100)	-0.009^{***}
	(0.002)
log(Distance (miles) to nearest city)	-0.128^{***}
	(0.027)
Water body less than 5 miles away	0.144^{***}
	(0.040)
log(Distance (miles) to nearest highway	-0.010
	(0.018)
Proportion of non-retail business acres	$-0.459^{\cdot}$
	(0.235)
Constant (intercept)	14.368^{***}
	(0.232)
F Statistic	114.318^{***}
df	8; 391
Observations	400
R^2	0.751
Adjusted R^2	0.746
Residual Std. Error	0.200

$\cdot$p<0.1; *p<0.05; **p<0.01; ***p<0.001

```
pollution_reg <- lm(pollutionIndex ~ log_distanceToHighway + nonRetailBusiness,
                    houseValue.2)
linearHypothesis(pollution_reg, "log_distanceToHighway")
```

```
## Linear hypothesis test
##
## Hypothesis:
## log_distanceToHighway = 0
##
## Model 1: restricted model
## Model 2: pollutionIndex ~ log_distanceToHighway + nonRetailBusiness
##
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    398 23396
## 2    397 20980  1    2415.7 45.712 4.912e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
linearHypothesis(pollution_reg, "nonRetailBusiness")
```

```
## Linear hypothesis test
##
```

```
## Hypothesis:
## nonRetailBusiness = 0
##
## Model 1: restricted model
## Model 2: pollutionIndex ~ log_distanceToHighway + nonRetailBusiness
##
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    398 35839
## 2    397 20980  1     14859 281.18 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Table 8: Regression model of pollutionIndex on the IVs

	Dependent variable:
	Pollution index (0-100)
log(Distance (miles) to nearest highway	3.431***
	(0.583)
Proportion of non-retail business acres	105.705***
	(8.693)
Constant (intercept)	22.379***
	(0.714)
F Statistic	446.481***
df	2; 397
Observations	400
R^2	0.624
Adjusted R^2	0.622
Residual Std. Error	7.270

·p<0.1; *p<0.05; **p<0.01; ***p<0.001

So let's perform a 2SLS regression using the (log of the) distance to the nearest highway and the proportion of acres dedicated to non-retail business as Instrumental Variables for the pollution index:

```
miv = ivreg(as.formula(paste("log_homeValue ~",
                             paste(names(stepwise.reg$coefficients)[-1],
                                   collapse = " + "), " | ",
                             paste(c(names(stepwise.reg$coefficients)[-c(1, 5)],
                                     "nonRetailBusiness",
                                     "log_distanceToHighway"),
                                   collapse = " + "))), data = houseValue.2)
```

Table 9: Regression model of log(homeValue)

	Dependent variable:
	log(Median price ($) of single-family house)
Percentage of low-income households	-0.024^{***}
	(0.003)
Average pupil-teacher ratio	-0.038^{***}
	(0.006)
Average number of bedrooms	0.097^{***}
	(0.027)
Pollution index (0-100)	-0.016^{***}
	(0.003)
log(Distance (miles) to nearest city)	-0.171^{***}
	(0.034)
Water body less than 5 miles away	0.156^{***}
	(0.039)
Constant (intercept)	14.784^{***}
	(0.300)
F Statistic	677.003^{***}
df	6; 393
Observations	400
R^2	0.741
Adjusted R^2	0.737
Residual Std. Error	0.203

·p<0.1; *p<0.05; **p<0.01; ***p<0.001

The other coefficients remaind almost unchanged (though not exactly; our aim was to eliminate the bias introduced by `pollutionIndex` because of its potential realtion with omitted variables), and now the impact of `pollutionIndex` has increased: an increase of one point decreases the median home value by 1.6% (and again, since the standard deviation of the Pollution Index is 11.8, an increase of 1 standard deviation in that index would decrease the value of the house by approximately 18.7%). The (robust) standard error has increased due to the use of IVs (not perfectly correlated with `pollutionIndex`) but it's still low enough so the coefficient is highly statistically significant.

Part 2

Modeling and Forecasting a Real-World Macroeconomic / Financial time series

Build a time-series model for the series in `lab3_series02.csv`, which is extracted from a real-world macroeconomic/financial time series, and use it to perform a 36-step ahead forecast. The periodicity of the series is purposely not provided. Possible models include AR, MA, ARMA, ARIMA, Seasonal ARIMA, GARCH, ARIMA-GARCH, or Seasonal ARIMA-GARCH models.

We start loading and inspecting the data:

```
financial <- read.csv('lab3_series02.csv', header = TRUE)
head(financial)
```

```
##   X DXCM.Close
## 1 1       9.88
## 2 2       9.79
## 3 3       9.68
## 4 4       9.64
## 5 5       9.42
## 6 6       9.47
```

```
# Check if 1st column is just an incremental index
all(financial$X == 1:dim(financial)[1])
```

```
## [1] TRUE
```

```
financial <- financial[, -1]
c(head(financial), tail(financial)) # 1st and last observations
```

```
##  [1]  9.88  9.79  9.68  9.64  9.42  9.47 67.63 70.49 67.79 68.72 68.43
## [12] 68.08
```

```
summary(financial)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.390   8.188  12.360  23.210  32.560 101.900
```

```
round(stat.desc(as.data.frame(financial), desc = TRUE, norm = TRUE), 2)
```

```
##              financial
## nbr.val        2332.00
## nbr.null          0.00
## nbr.na            0.00
## min               1.39
## max             101.91
## range           100.52
## sum           54125.73
## median           12.36
## mean             23.21
## SE.mean              0.49
## CI.mean.0.95         0.95
## var                549.61
## std.dev             23.44
## coef.var             1.01
## skewness             1.54
## skew.2SE            15.21
## kurtosis             1.24
## kurt.2SE             6.11
## normtest.W           0.75
## normtest.p           0.00
```

The dataset contains 2332 observations, with no dates (there was another column but that just contains an incremental index that adds no information).

The histogram is very right-skewed. Anyway, it is not informative in time series (it tells us nothing about their dynamics).

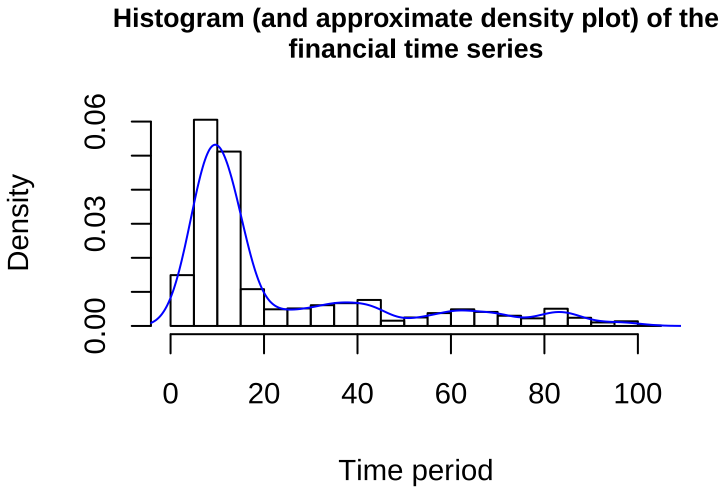


Figure 16: Histogram (and approximate density plot) of the values of the financial time series

If we plot **the time series** (see next page) we observe that it **is quite persistent, it is difficult to observe any seasonality, and the variance seems to increase**. The ACF and PACF (plotted after the time series, also in the next page) resemble a random walk or an AR(1) model very much: the ACF decreases very slowly, and the PACF falls sharply after the 1st lag.

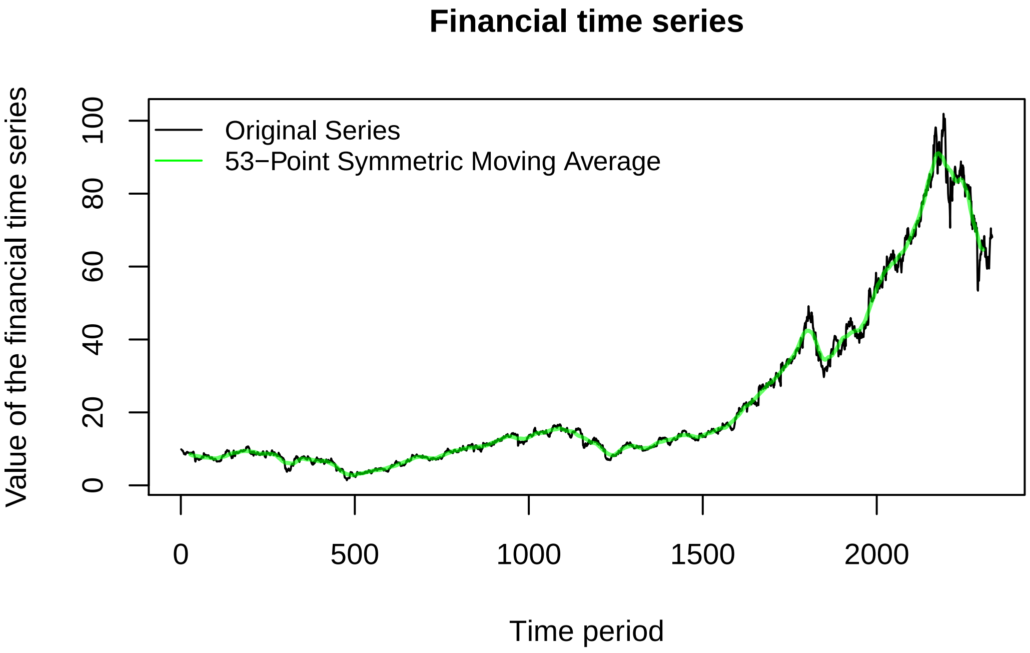


Figure 17: Time series plot of the financial series

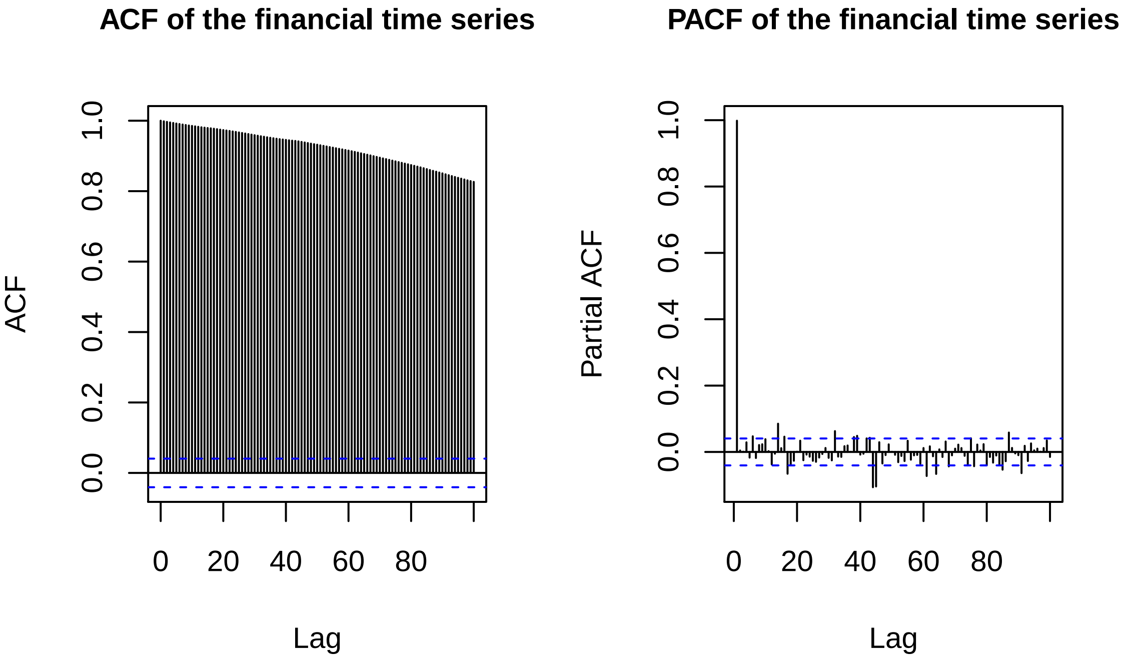


Figure 18: ACF and PACF of the financial time series

The PACF at the 16th lag is also significant, which might make us think that that's the seasonality, but we can check that quickly, by analyzing the ACF of the differenced series:

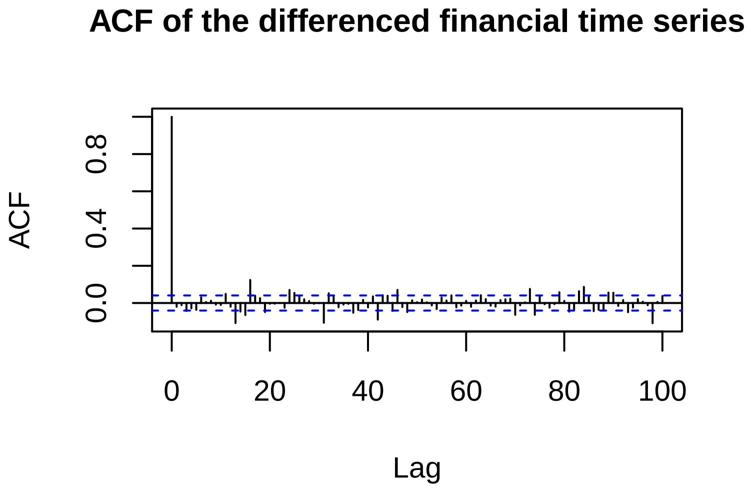


Figure 19: ACF of the differenced financial time series

For a (strong) seasonal component, the ACF of the differenced series would be significant at the corresponding lag (16 in this case)… and also at multiples of it (32, 48, …), which is not the case (besides, financial series do not usually have a strong seasonal component as other series do).

Since this is a **financial** series, let's analyze the return (or relative increment), defined as:

$$r_t = \frac{x_t - x_{t-1}}{x_{t-1}}$$

If we define y_t as $y_t = \log(x_t)$ then for small increases of x_t (i.e., if $x_t/x_{t-1} \approx 1 \; \forall t\}$), we can use the following approximation:

$$(\mathbf{1}-\mathbf{B})\log(\mathbf{x_t}) = \log(\mathbf{x_t}) - \log(\mathbf{x_{t-1}}) = \log\left(\frac{x_t}{x_{t-1}}\right) = \Delta\log(x_t) \approx \frac{x_t}{x_{t-1}} - 1 = \mathbf{r_t}$$

I.e., if we difference the log of the series we have a new series close to the return of the original one.

```
ret <- diff(financial) / financial[2:length(financial)]
diff_log <- diff(log(financial))
tail(cbind(ret, diff_log))
```

```
##                  ret     diff_log
## [2326,] -0.004140174 -0.004131628
## [2327,]  0.040573131  0.041419184
## [2328,] -0.039828883 -0.039056164
## [2329,]  0.013533178  0.013625586
## [2330,] -0.004237907 -0.004228953
## [2331,] -0.005141011 -0.005127841
```

```
head(cbind(ret, diff_log))
```

```
##                ret     diff_log
## [1,] -0.009193054 -0.009151055
## [2,] -0.011363636 -0.011299555
## [3,] -0.004149378 -0.004140793
## [4,] -0.023354565 -0.023086020
## [5,]  0.005279831  0.005293819
## [6,] -0.033842795 -0.033282729
```

Let's now plot the log and difference of logs (i.e., the log return) of the financial series, as well as the respective ACFs and PACFs:

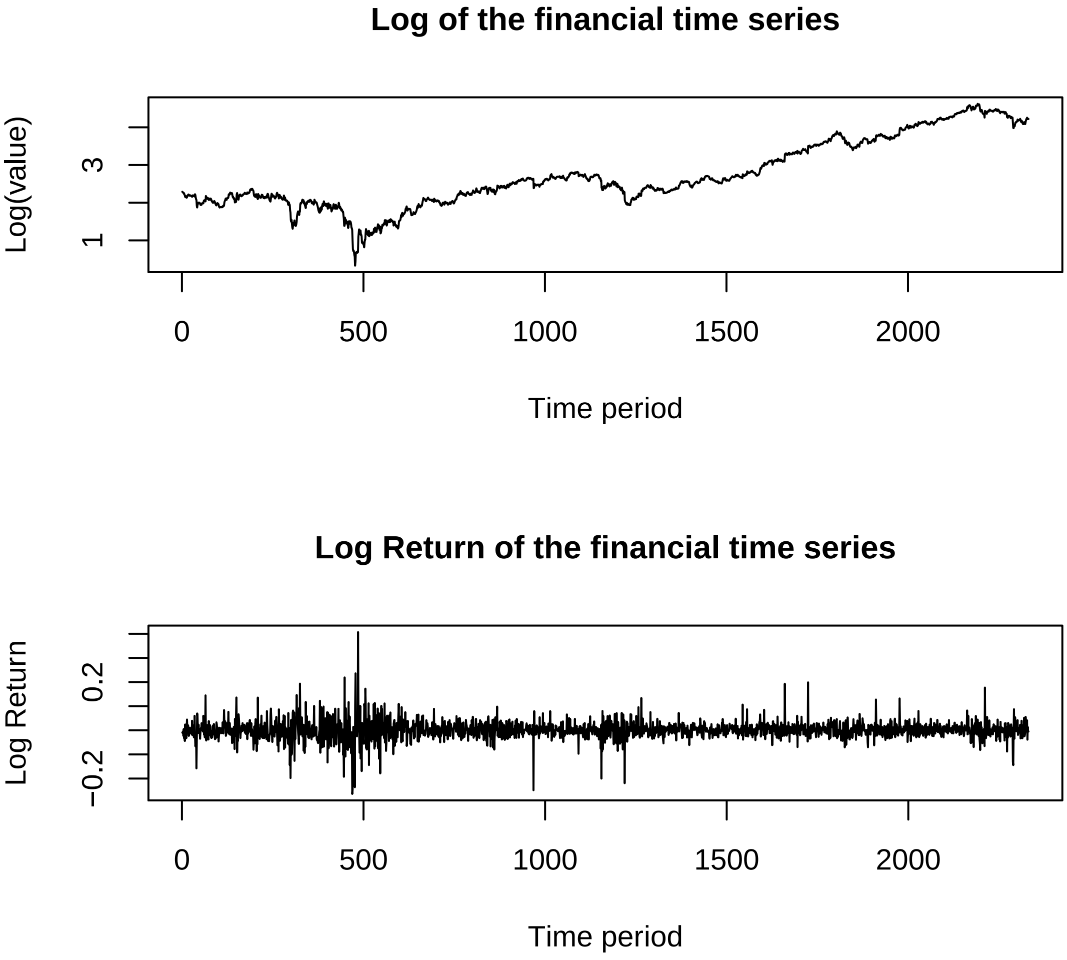


Figure 20: Time series plot of the financial series

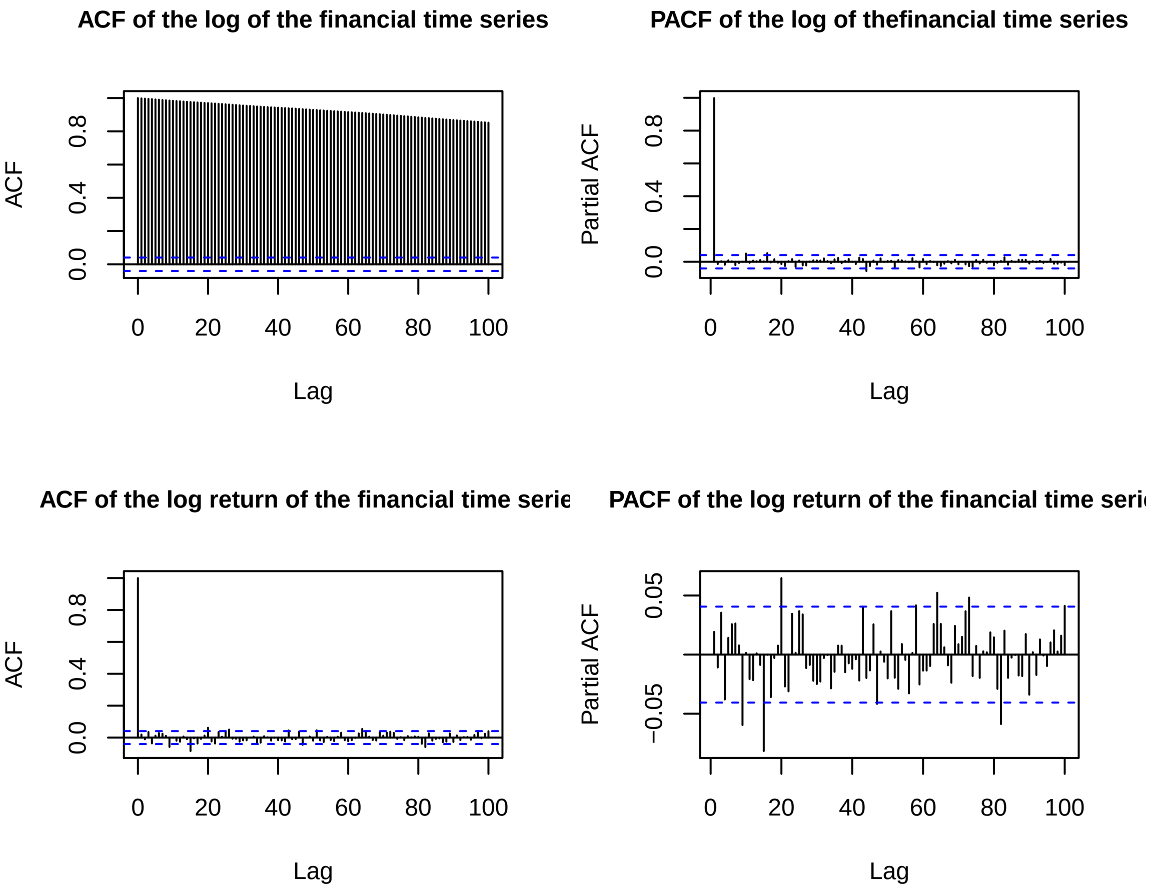


Figure 21: ACF and PACF of the log return of the financial time series

Both the time series plot and the ACF and PACF of the log of the series resemble a **random walk**. Similarly (and consequently), the difference resemble a **white noise**. . . with one caveat: **the variance is not constant** (but higher around the 500th observation).

So a good model for the log of the series would be simply an ARIMA(0,1,0)! We can predict using such model and exponentiate to get the predictions for the original series (those predictions should be corrected because the model is optimized for the log, not for the original series, and since the logarithm is not a linear operation, $\exp(E[\log(x_t)]) \neq E[\exp(\log(x_t))] = E[x_t]$).

```r
arima010.fit <- Arima(log(financial), order = c(0, 1, 0))
arima010.fit.fcast <- forecast.Arima(arima010.fit, h = 36, level = .95)
# NO NEED TO APPLY EXP(): Arima() ADMITS A BOX-COX TRANSFORMATION
# WHICH IS EQUAL TO LOG WHEN LAMBDA = 0
arima010.fit2 <- Arima(financial, order=c(0, 1, 0), lambda = 0)
arima010.fit.fcast2 <- forecast.Arima(arima010.fit2, h = 36)
```

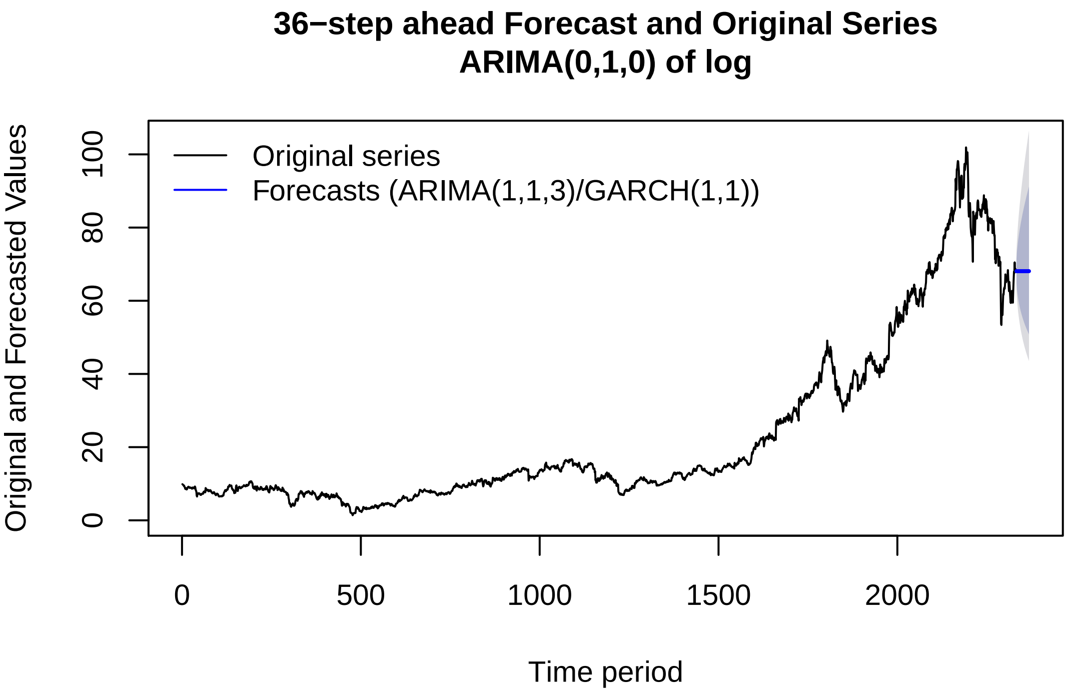


Figure 22: 36-step ahead forecasts of the financial time series based on an ARIMA(0,1,0) model fitted to the log of the data

But as we mentioned before, the variance of the residuals might be not constant. An inspection of the square of the residuals of the integrated model (see the 1st Figure in the following page) confirms that, so we should enhance the confidence intervals of our forecasts (their mean value will remain the same) with a GARCH model.

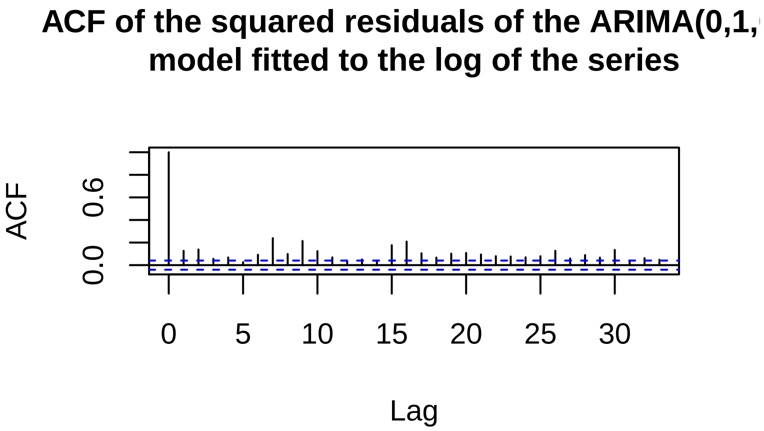


Figure 23: ACF of the squared residuals of the ARIMA(0,1,0) model fitted to the log of the financial series

Let’s try a GARCH(1,1) model:

```
financial.garch <- garch(resid(arima010.fit), trace = FALSE)
```

The residuals of such model resemble a white noise (with constant variance!), so the ARIMA(0,1,0)/GARCH(1,1) of the log is a good fit.

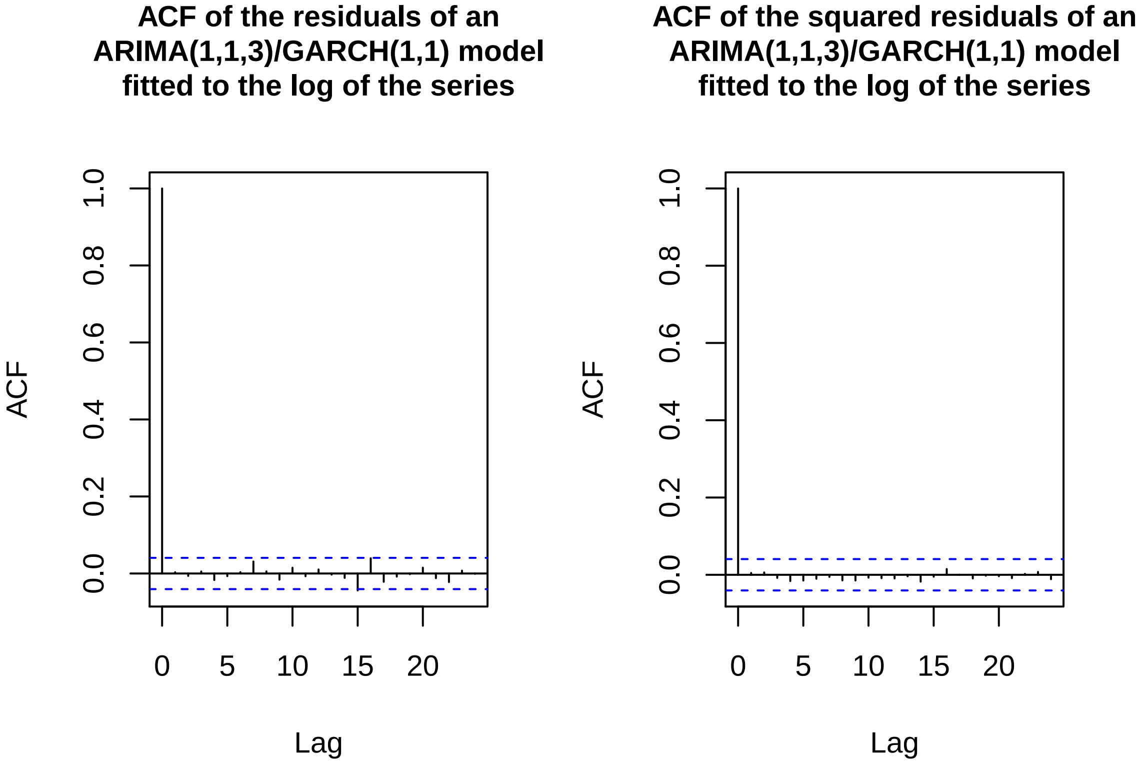


Figure 24: ACF of the residuals and squared residuals of an ARIMA(1,1,3)/GARCH(1,1) model fitted to the U.S. inflation-adjusted average gas prices

Thus, we can use this enhanced model to get narrower confidence intervals of the previous forecasts. In a SARIMA model, the 95% **prediction intervals** for x_{n+h} (where n is the last observation and h is the number of steps ahead) satisfies:

$$\Pr\left(|f_{n,h} - x_{n+h}| < 1.96\sqrt{Var(e_{n,h})}\right) = 0.95$$

where $f_{n,h}$ is the forecast, $e_{n,h}$ is the prediction error, and:

$$Var(e_{n,h}) = \sigma^2 \sum_{j=0}^{h-1} \Psi_j^2$$

σ^2 is the—supposedly constant—variance of the noise, which we approximate by the residuals, and Psi_j are the coefficients of $\Psi(B) = \Phi(B)/\Theta(B) = x_t/\epsilon_t$. **The sum of h terms is what makes the prediction interval gets wider over time**.

When the series are conditional heteroskedastic, we have to **substitute that constant variance ($\sigma^2 =$ `var(resid(model.fit))`) by the variance (that changes with time) given by the GARCH model**.

```
financial.garch11 <- garchFit(~ garch(1,1), data = resid(arima010.fit),
                              trace = FALSE)
res.fcst <- predict(financial.garch11, n.ahead = 36, conf = .95)
```

```
# Compare the previous std. dev. with the (changing) new one
sd(resid(arima010.fit))
```

```
## [1] 0.03802252
```

```
c(head(res.fcst$standardDeviation), tail(res.fcst$standardDeviation))
```

```
##  [1] 0.03067630 0.03102770 0.03135790 0.03166841 0.03196062 0.03223578
##  [7] 0.03580214 0.03586717 0.03592885 0.03598734 0.03604283 0.03609546
```

```
# Add the mean prediction of GARCH (close to zero) to the prediction of SARIMA
# and subtract/add the previous CI / sigma * sigma_t
fcst.lower <- exp(arima010.fit.fcast$mean + res.fcst$meanForecast -
                    c(arima010.fit.fcast$upper - arima010.fit.fcast$mean) /
                    sd(resid(arima010.fit)) * res.fcst$standardDeviation)
fcst.upper <- exp(arima010.fit.fcast$mean + res.fcst$meanForecast +
                    c(arima010.fit.fcast$upper - arima010.fit.fcast$mean) /
                    sd(resid(arima010.fit)) * res.fcst$standardDeviation)
```

As shown below, the new 95% confidence intervals are almost (slightly narrower) than the previous ones (without applying the GARCH model).

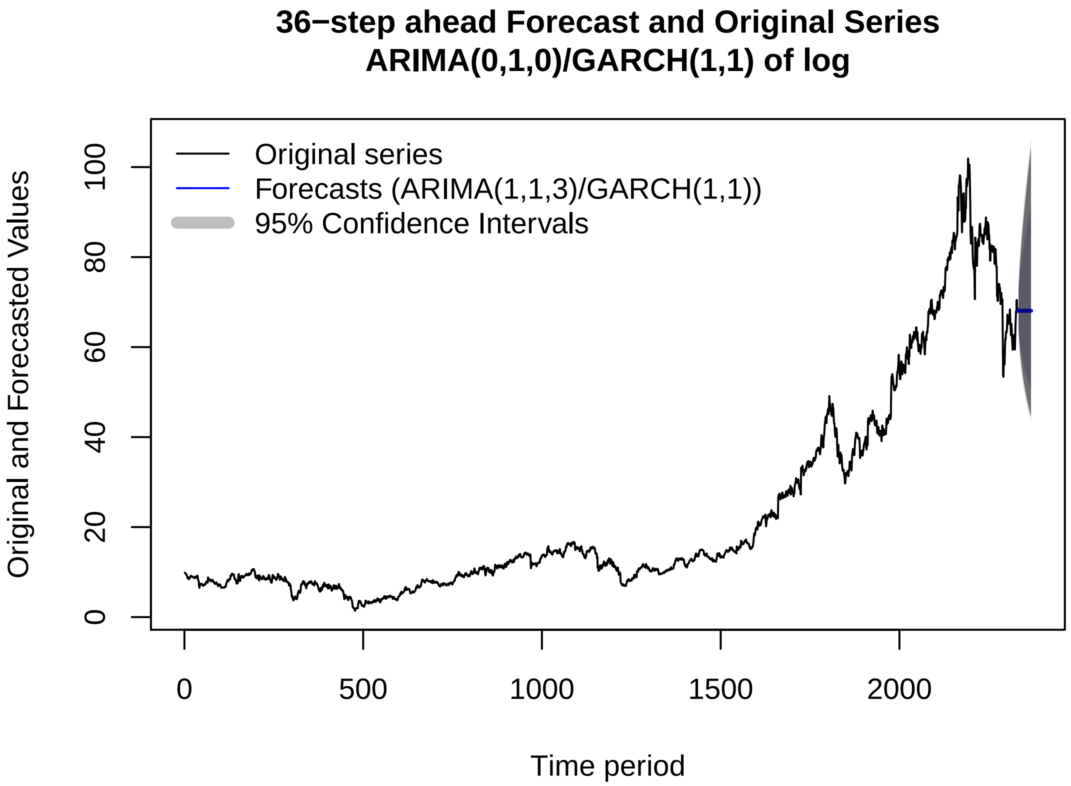


Figure 25: 36-step ahead forecasts of the financial time series based on an ARIMA(0,1,0)/GARCH(1,1) model fitted to the log of the data

For a self-made version of GARCH forecasting (not using `fGarch`), see Part 4.

Part 3

Forecast the Web Search Activity for global Warming

Imagine that you group is part of a data science team in an apparel company. One of its recent products is Global-Warming T-shirts. The marketing director expects that the demand for the t-shirts tends to increase when global warming issues are reported in the news. As such, the director asks your group to forecast the level of interest in global warming in the news. The dataset given to your group captures the relative web search activity for the phrase, "global warming" over time. For the purpose of this exercise, ignore the units reported in the data as they are unimportant and irrelevant. Your task is to produce the weekly forecast for the *next 3 months* for the relative web search activity for global warming. For the purpose of this exercise, treat it as a *12-step ahead forecast.*

The dataset for this exercise is provided in `globalWarming.csv`. Use only models and techniques covered in the course (up to lecture 13). Note that one of the modeling issues you may have to consider is whether or not to use the entire series provided in the data set. Your choice will have to be clearly explained and supported with empirical evidence. As in other parts of the lab, the general instructions in the *Instruction Section* apply.

We start loading the data and inspecting (and transforming[1]) the resulting dataframe.

```r
GW <- read.csv('globalWarming.csv', header = TRUE)
rbind(head(GW,4 ), tail(GW, 4))
```

```
##        Date data.science
## 1    1/4/04       -0.440
## 2   1/11/04       -0.474
## 3   1/18/04       -0.423
## 4   1/25/04       -0.551
## 627  1/3/16        3.662
## 628 1/10/16        3.721
## 629 1/17/16        4.087
## 630 1/24/16        4.104
```

```r
GW$Date <- as.Date(as.character(GW$Date), '%m/%d/%y')
# Day of week of 1st observation
as.character(wday(GW$Date[1], label = TRUE, abbr = FALSE))
```

```
## [1] "Sunday"
```

```r
# Check that all observations correspond to same day of the week
all(wday(GW$Date, label = TRUE) == wday(GW$Date[1], label = TRUE))
```

```
## [1] TRUE
```

```r
# Check that all weeks between start and end dates appear in the dataset
identical(GW$Date, seq(min(GW$Date), max(GW$Date), by=7))
```

```
## [1] TRUE
```

[1] Converting the dates from factors to dates, shortening the names of the dataframe and variables, etc.

```
names(GW)[2] <- "DS"
summary(GW)
```

```
##       Date                  DS
##  Min.   :2004-01-04   Min.   :-0.551000
##  1st Qu.:2007-01-08   1st Qu.:-0.506000
##  Median :2010-01-13   Median :-0.485000
##  Mean   :2010-01-13   Mean   : 0.000038
##  3rd Qu.:2013-01-18   3rd Qu.:-0.200000
##  Max.   :2016-01-24   Max.   : 4.104000
```

```
round(stat.desc(as.data.frame(GW$DS), desc = TRUE, norm = TRUE), 2)
```

```
##                 GW$DS
## nbr.val        630.00
## nbr.null         0.00
## nbr.na           0.00
## min             -0.55
## max              4.10
## range            4.66
## sum              0.02
## median          -0.48
## mean             0.00
## SE.mean          0.04
## CI.mean.0.95     0.08
## var              1.00
## std.dev          1.00
## coef.var     26249.94
## skewness         2.12
## skew.2SE        10.88
## kurtosis         3.47
## kurt.2SE         8.93
## normtest.W       0.59
## normtest.p       0.00
```

```
# Create a time series object (weekly observations)
GW.ts <- ts(GW$DS, start = 2004 + day(min(GW$Date)) / 365.25,
            freq = 365.25 / 7)
```

The dataset contains 630 observations of a numeric variable (`data.science`, which we shortened to `DS`), from 2004-01-04 to 2016-01-24 (i.e., approximately 12.1 years). All dates correspond to Sundays (so we have weekly observations), and there are no missing values for any Sunday, and all Sundays between the start and end date are available in the dataset. The numeric variable must have been standardized (i.e., rescaled by subtracting its mean and dividing by its standard deviation), and that's the possible reason it has zero mean and unit variance. In any case, it does not resemble a normal distribution at all, as it is very right-skewed.

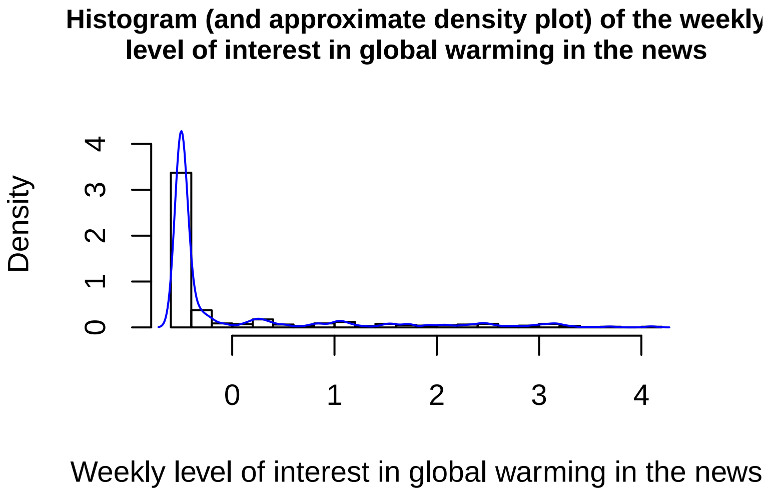


Figure 26: Histogram (and approximate density plot) of the weekly level of interest in global warming in the news

But the density distribution of a time series tells us nothing about its dynamics: we have to look at the time series plot to know that the value of the (standardized) variable was almost flat until 2012, when it started growing almost linearly (with some shocks up and down, the most prominent ones a fall at the end of 2015 and a peak right after that).

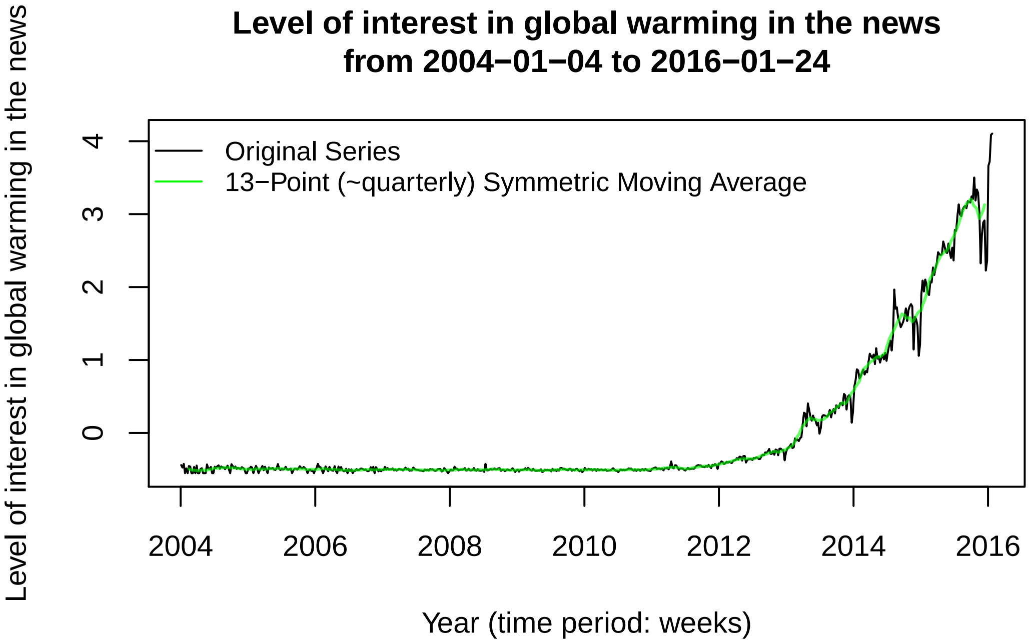
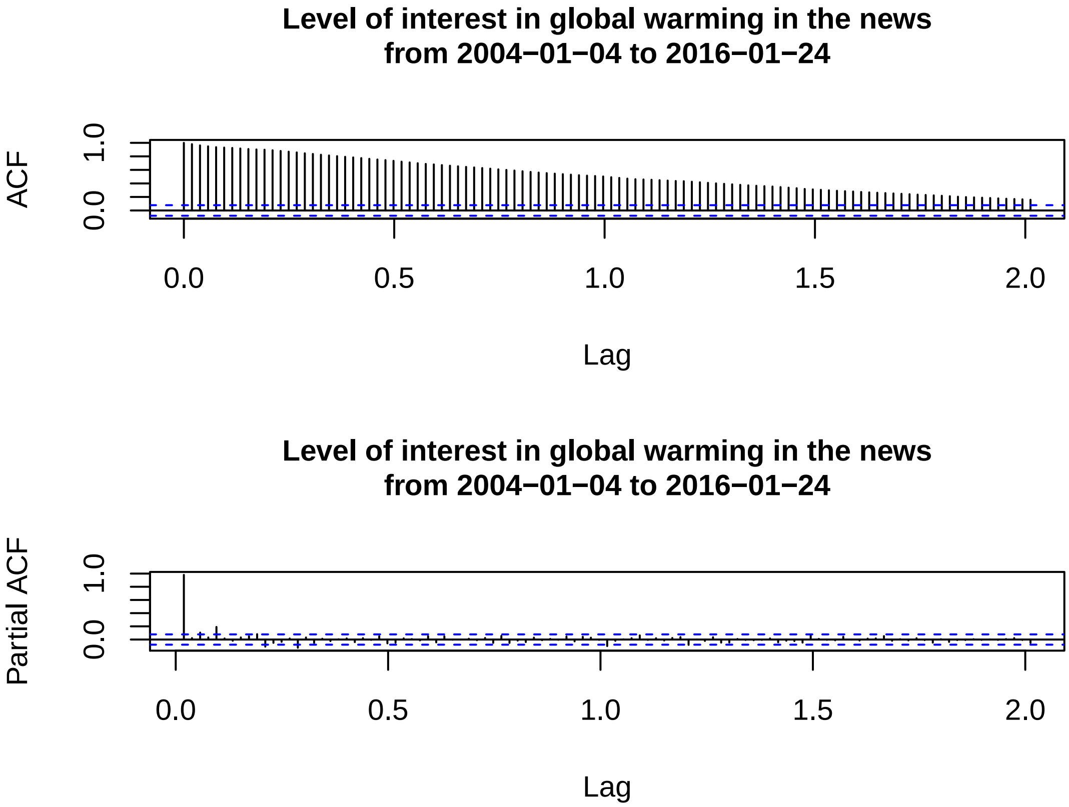


Figure 27: Level of interest in global warming in the news from 2004-01-04 to 2016-01-24

The ACF and PACF (see next page) might make us think of a simple AR(1) model: the ACF decreases very slowly and the PACF falls sharply after the 1st lag. But the PACF is also significant at the 5th lag, so more complex models may be necessary.


Level of interest in global warming in the news from 2004−01−04 to 2016−01−24

Level of interest in global warming in the news from 2004−01−04 to 2016−01−24

Figure 28: ACF and PACF of th level of interest in global warming in the news from 2013-03-17 to 2016-01-24

Next **we should focus on what data to use to forecast the next 3 months: the whole dataset or the last observations)**. The time series plot makes us think that the process that generates the series might not be the same all the time: it is unlikely that the same process, which generated values close to zero and small variance for several years, then started to generate increasing (and more variable) values. It is more plausible that some event (likely a higher level of awareness about global warming) changed the process, making it different. Hence, **modelling two subsequent processes in the same way may lead to wrong results**.

Let's begin decomposing the time series:

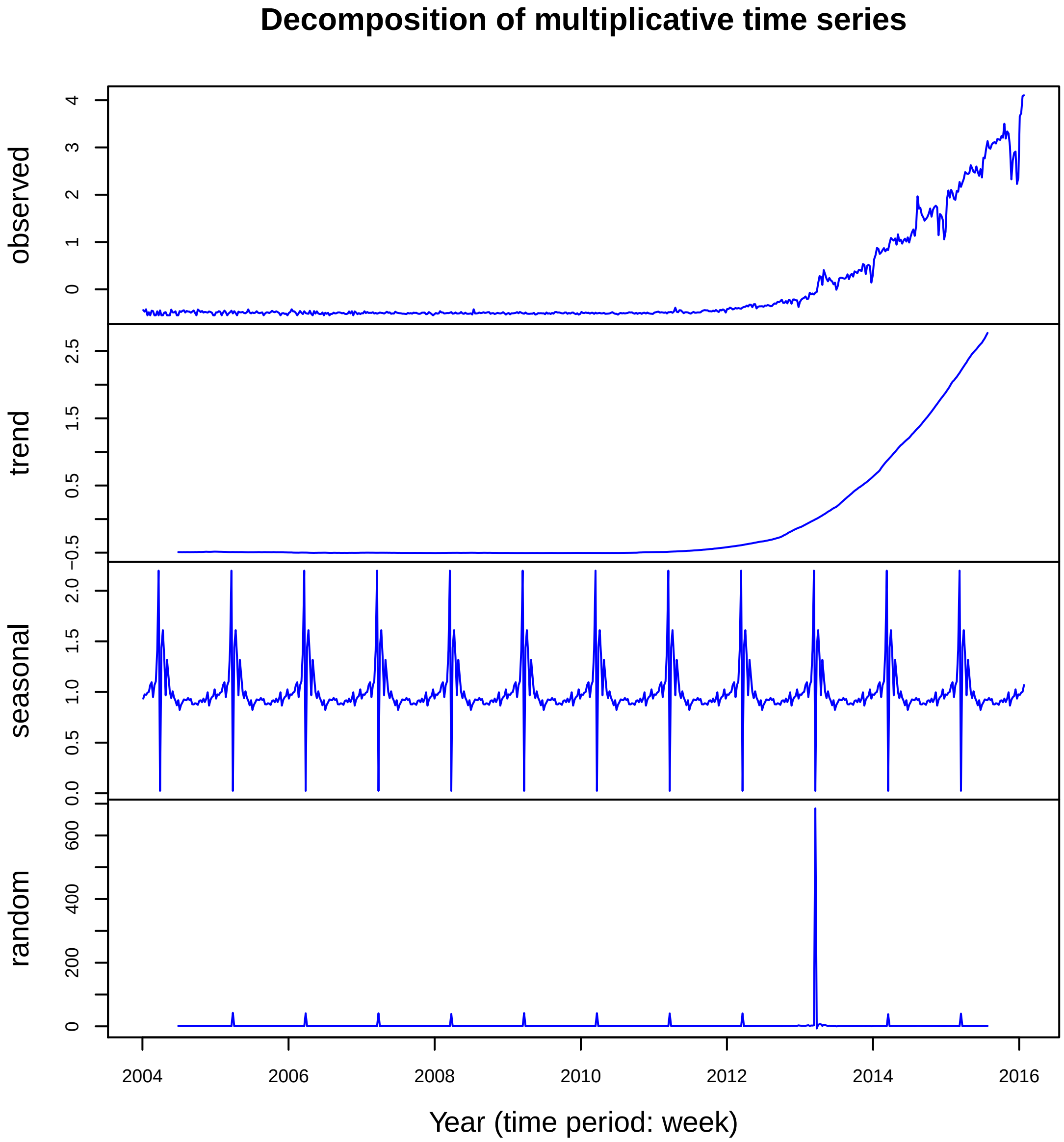


Figure 29: Multiplicative decomposition of the time series of the level of interest in global warming in the news from 2004-01-04 to 2016-01-24

The **multiplicative decomposition** (much more informative than the additive one, that's why we don't plot the latter) shows a shock at the beginning at 2013 (maybe some shocking news that raised interest in global warming?) that resulted in the increasing trend from that date on. It also shows a high yearly seasonal component. Let's check the exact date:

```
(shock.position <- which(decompose(GW.ts,
                                   type = 'multiplicative')[['random']] ==
                         max(decompose(GW.ts,
                                       type = 'multiplicative')[['random']],
                             na.rm = TRUE))) # 481
```

```
## [1] 481
```

```
(shock.date <- GW$Date[shock.position]) # "2013-03-17"
```

```
## [1] "2013-03-17"
```

Of course that may not be the exact date (just the estimate from the decomposition), but we'll use it as a potential start for a reduced dataset.

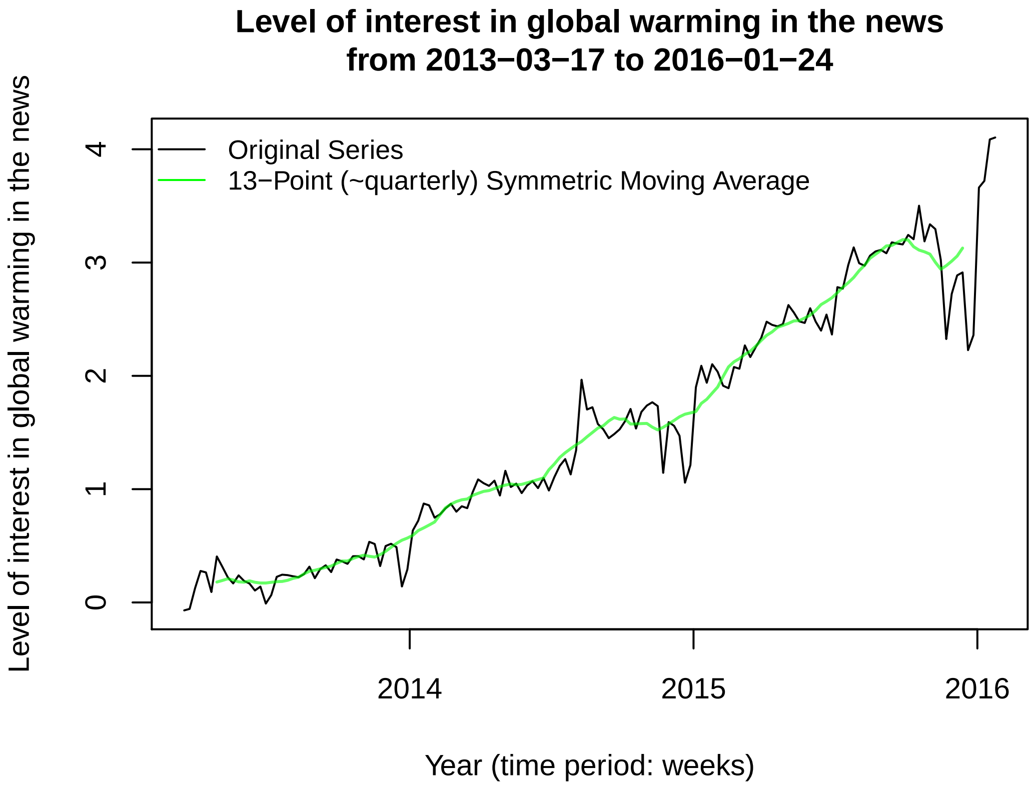


Figure 30: Level of interest in global warming in the news from 2013-03-17 to 2016-01-24

To check which approach is more appropriate, we consider two datasets: one with all the data we have, and another with only the last observations (from that date in 2013 on, when the level of interest in global warming in the news started growing).

```
# Whole dataset
GW.whole <- GW.ts
# Reduced dataset: last obsservations (from shock date)
GW.last <- window(GW.whole, start = year(shock.date) + (as.numeric(difftime(
  shock.date, as.Date(paste0(year(shock.date), "-1-1")))) + 1) / 365.25,
  freq = 365.25/7)
# Fit the "best" ARIMA model for the whole dataset
(arima.whole.fit <- auto.arima(GW.whole, seasonal = TRUE))
```

```
## Series: GW.whole
## ARIMA(1,1,1)(0,1,1)[52]
##
## Coefficients:
##          ar1      ma1     sma1
##       0.4195  -0.7714  -0.2117
## s.e.  0.0890   0.0680   0.0471
##
## sigma^2 estimated as 0.007879:  log likelihood=578.89
## AIC=-1149.78   AICc=-1149.71   BIC=-1132.35
```

Looking at the ACFs, the residuals of both models resemble a white noise slightly well. But when it comes to the PACFs, only the 2nd model, fitted to the reduced dataset, really resembles a white noise.

```
# Fit the "best" ARIMA model for the reduced dataset
(arima.last.fit <- auto.arima(GW.last, seasonal = TRUE))
```

```
## Series: GW.last
## ARIMA(0,1,1)(0,1,0)[52]
##
## Coefficients:
##           ma1
##       -0.4017
## s.e.   0.1090
##
## sigma^2 estimated as 0.03831:  log likelihood=21.07
## AIC=-38.14   AICc=-38.02   BIC=-32.99
```

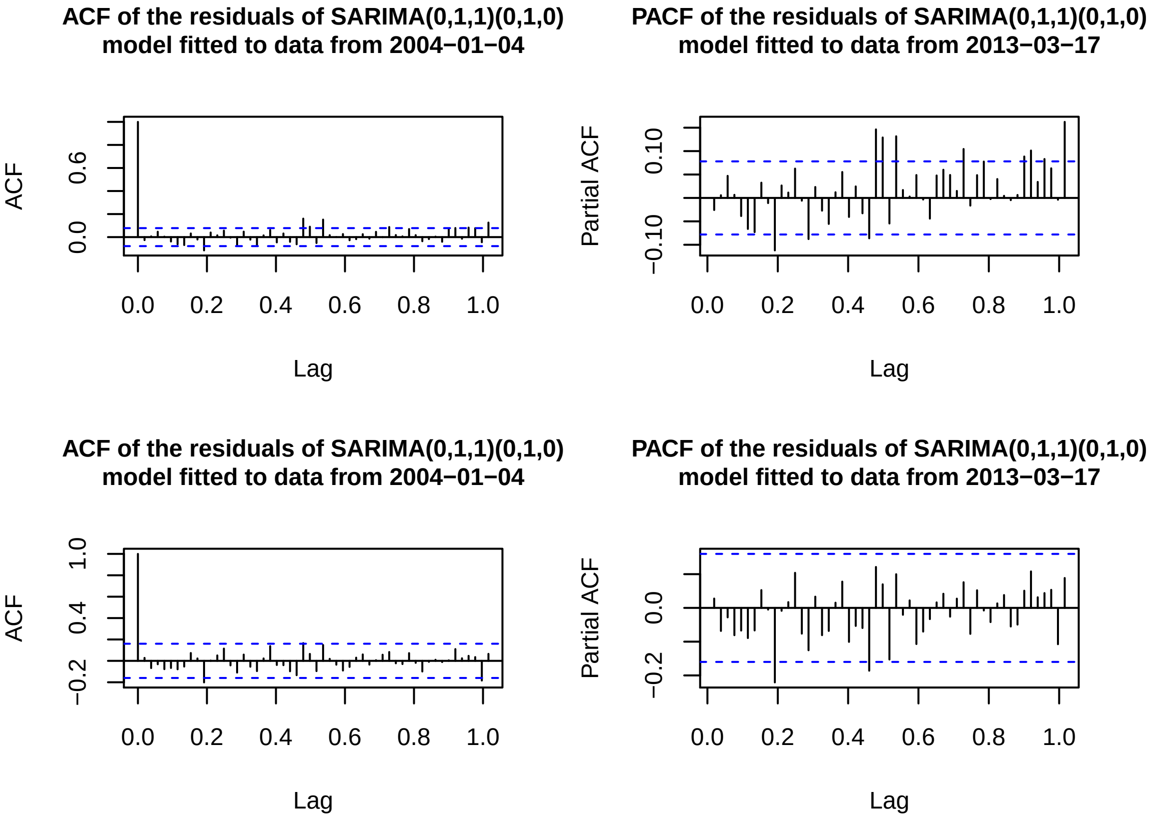


Figure 31: ACF and PACF of residuals of the two SARIMA fitted to the level of interest in global warming in the news from 2004 and 2013, respectively

To have a better idea of what dataset (complete or limited to last observations) leads to a better model, we now conduct an out-of-sample fit testing with the last 15 observations (the reduced time series contains 150 observations):

```r
# Out-of-sample fit of both models
# Training sets (exclude last 15 observations, 10% of the reduced dataset)
GW.whole.train <- window(GW.whole, start = time(GW.whole)[1],
                         end = time(GW.whole)[length(GW.whole)-15])
GW.last.train <- window(GW.last, start = time(GW.last)[1],
                        end = time(GW.last)[length(GW.last)-15])
# Test set
GW.test <- window(GW.whole, start = time(GW.whole)[length(GW.whole)-15+1],
                  end = time(GW.whole)[length(GW.whole)])
# Fit new models for the training sets using same coefficients
arima.whole.oos.fit <- Arima(GW.whole.train,
                             order = arima.whole.fit$arma[c(1, 6, 2)],
                             seas = list(order = arima.whole.fit$arma[c(3, 7,
                                                                        4)],
                                         freq = arima.whole.fit$arma[5]))
arima.last.oos.fit <- Arima(GW.last.train,
                            order = arima.last.fit$arma[c(1, 6, 2)],
                            seas = list(order = arima.last.fit$arma[c(3, 7,
                                                                      4)],
                                        freq = arima.last.fit$arma[5]))
# Predict next 15 observations based on each model
arima.whole.oos.fit.fcast <- forecast.Arima(arima.whole.oos.fit, h = 15)
arima.last.oos.fit.fcast <- forecast.Arima(arima.last.oos.fit, h = 15)
```

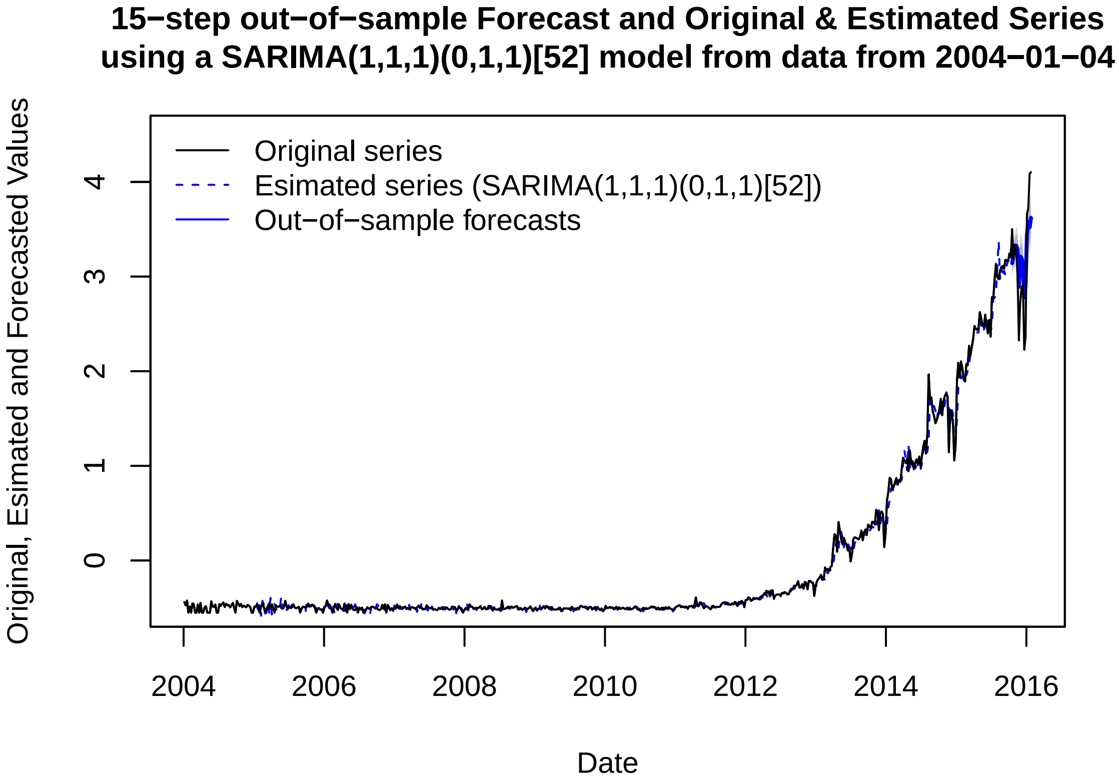


Figure 32: Out-of-sample fit of the SARIMA(1,1,1)(0,1,1)[52] model to the level of interest in global warming in the news from 2004

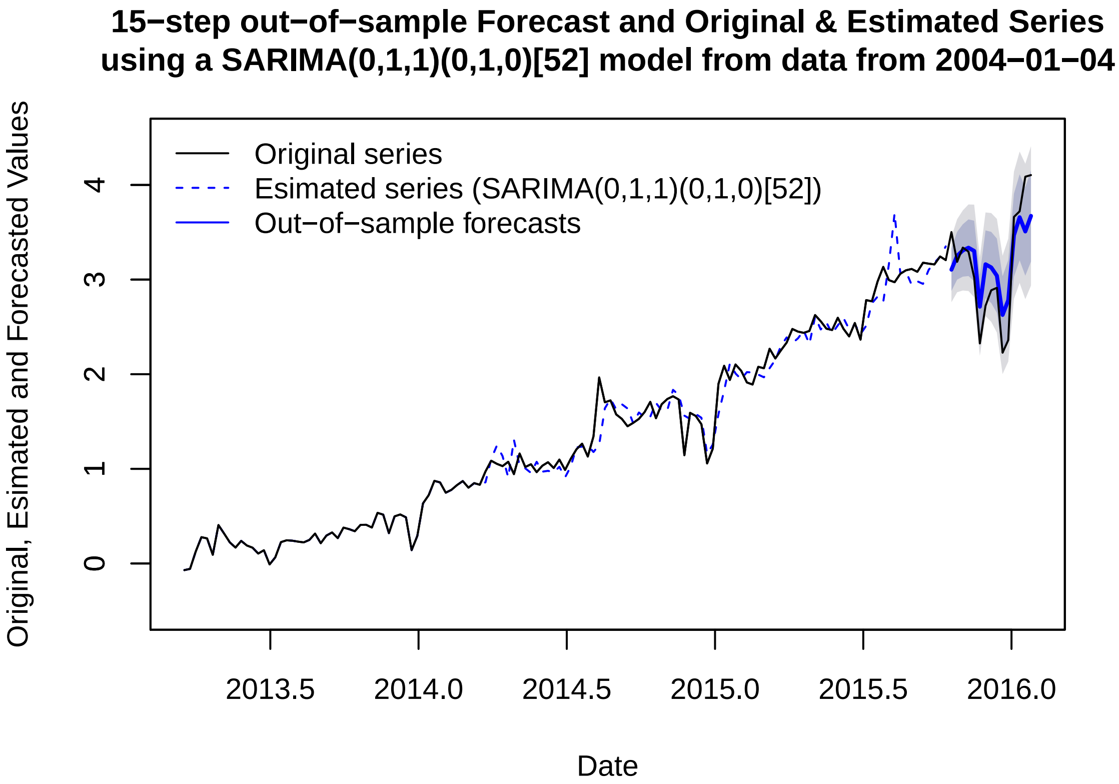


Figure 33: Out-of-sample fit of the SARIMA(1,1,1)(0,1,1)[52] model to the level of interest in global warming in the news from 2004

If we combine last 2 Figures and zoom in (see the Figure in the following page), we confirm that, though the mean forecast is very similar for both models, the SARIMA that was constructed using only data from 2013-03-17 on is a much better fit: at least the original values always fall within the confidence region of the forecasts, which never happens for the SARIMA model constructed from the complete series (because it's narrower). So we have justified the use of a reduced version of the series, containing only observations from last years, rather than the entire series. Now we just to have to build the definitive model and predict the 12-step ahead forecasts.

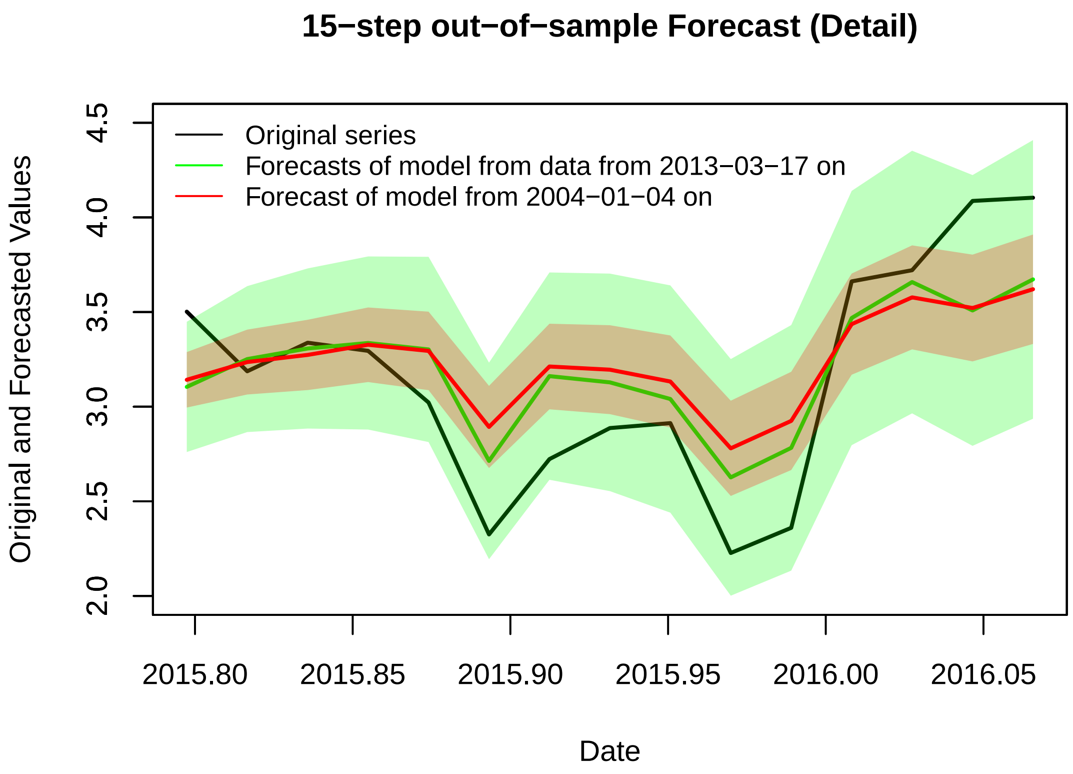


Figure 34: Detail of the out-of-sample fit of the SARIMA models fitted to the complete and reduced version of the time series

Actually, **we already have a model (SARIMA(0,1,1)(0,1,0)[52]) fitted to the reduced series, which**:

- **has the lowest AIC value (that's the criteria followed by `auto.arima()`)**,
- **captures the seasonality**,
- **has residuals that resemble white noise**,
- **has a good out-of-sample**, and
- **it's relatively simple** (an MA component, and 1 seasonal and 1 non-seasonal difference).

That model would correspond to:

$$(1-B^s)(1-B)x_t = \Phi_1(B)\omega_t = (1+\phi_1 B)\omega_t$$

$$(1-B^52)(1-B)x_t = (1-0.402B)\omega_t$$

$$x_t = x_{t-1} + x_{t-52} - x_{t-53} + \omega_t - 0.402\omega_{t-3}$$

where $\{w_t\}$ is a white noise series with mean zero and variance σ^2 (and, of course, $\{x_t\}$ is the level of interest in global warming in the news since 2013-03-17).

We just need to check whether the residuals of that SARIMA model are conditional heteroskedastic.

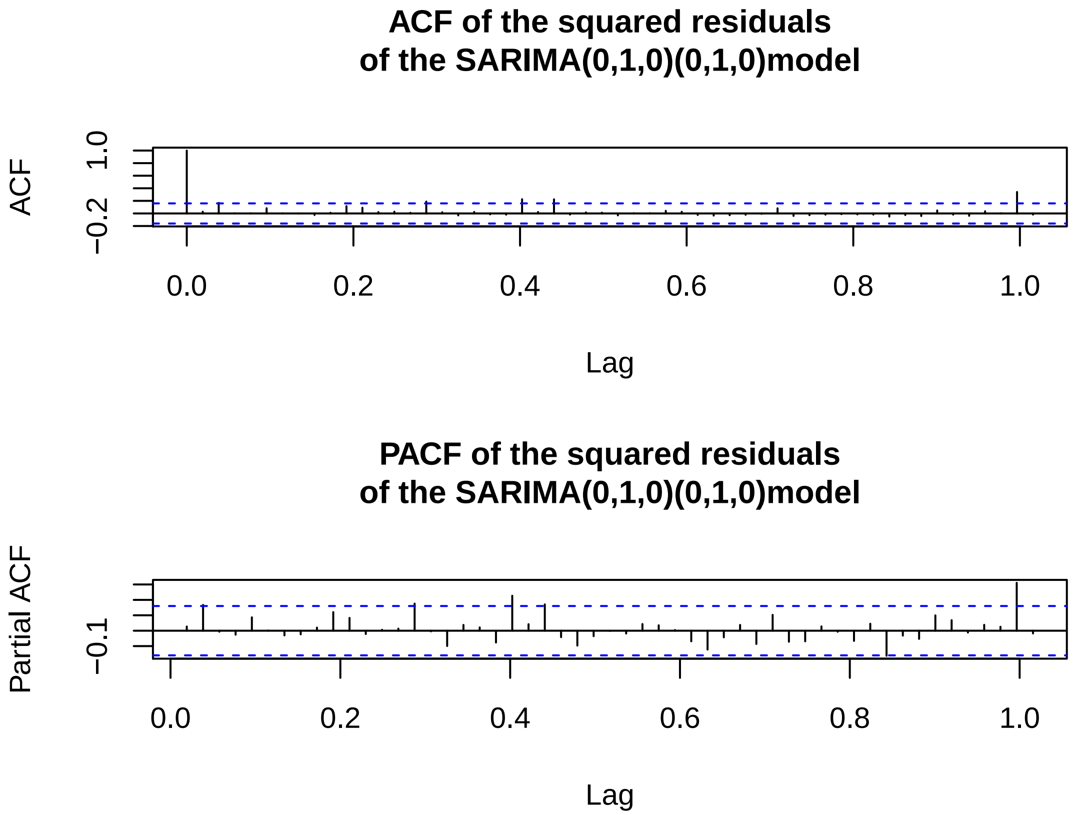


Figure 35: ACF and PACF of the squared residuals of the SARIMA(0,1,1)(0,1,0) model fitted to the level of interest in global warming in the news since 2013-03-17

The ACF and PACF of the squared residuals resemble quite reasonably a white noise (only very few of them are significant at some lags), so there's no need to complement our model with a GARCH model (that would enhance the confidence intervals of our predictions).

```
arima.last.fit.fcast <- forecast.Arima(arima.last.fit, h = 12)
```

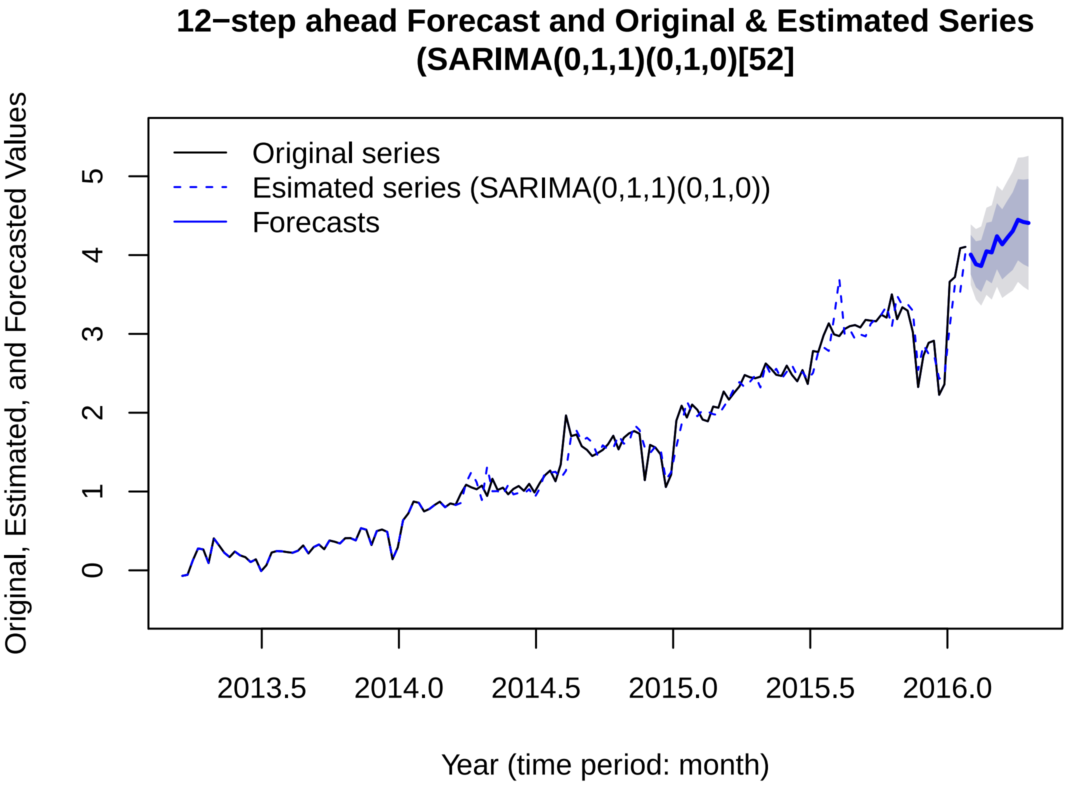


Figure 36: 12-step ahead forecasts (from 2016-01-31 to 2016-04-17) of the (standardized?) level of interest in global warming in the news based on a SARIMA(0,1,1)(0,1,0) model fitted to weekly data from 2013-03-17 to 2016-01-24

Part 4

Forecast Inflation-Adjusted Gas Price

During 2013 amid high gas prices, the Associated Press (AP) published an article about the U.S. inflation-adjusted price of gasoline and U.S. oil production. The article claims that there is "*evidence of no statistical correlation*" between oil production and gas prices. The data was not made publicly available, but comparable data was created using data from the Energy Information Administration. The workspace and data frame `gasOil.Rdata` contains the U.S. oil production (in millions of barrels of oil) and the inflation-adjusted average gas prices (in dollars) over the date range the article indicates.

In support of their conclusion, the AP reported a single p-value. You have two tasks for this exericse, and both tasks need the use of the data set `gasOil.Rdata`.

1st task

Your first task is to recreate the analysis that the AP likely used to reach their conclusion. Thoroughly discuss all of the errors the AP made in their analysis and conclusion.

It would seem reasonable (and that's probably what interested parties tell about the benefits of drilling) that the more oil is produced in the U.S. (mainly because of that new technique), the lower the price of gasoline would be. In other words, we might expect a highly significant **negative** correlation between domestic oil production and (inflation-adjusted) prices[2].

Possible sources for the mentioned article might be this one or this other one (though none of them reproduced the phrase "*evidence of no statistical correlation*"). That phrase would be the first error made by the AP (in case they used it): in **hypothesis testing**, we can talk of *no evidence of correlation* (or any other fact) but not of *evidence of no correlation* (in general, of evidence of a fact not occurring). Remember that, whatever a **null hypothesis** is (in this case, that the **correlation** is—**not statistically significantly different from**—**zero**), we can never prove or confirm it, just claim that we have evidence or not to reject it. To put an example, if we toss a coin N times and get heads N times, we do not have evidence that the coin is fair (i.e., $\Pr(heads) = 0.5$); but we should not claim that we have evidence that the opposite is true (i.e., the coin is unfair or biased); the most we can say, strictly speaking, is that we are quite confident (the more confident the greater the number of tosses).

Let's continue by loading and exploring the data frame we are given:

```r
load('gasOil.Rdata')
rbind(head(gasOil,4 ), tail(gasOil, 4))
```

```
##           Date Production    Price
## 1   1978-01-01    259.150 2.456692
## 2   1978-02-01    234.544 2.441220
## 3   1978-03-01    270.324 2.425818
## 4   1978-04-01    264.526 2.414277
## 407 2011-11-01    179.099 3.540914
## 408 2011-12-01    185.712 3.417614
## 409 2012-01-01    190.358 3.527641
## 410 2012-02-01    180.969 3.726987
```

[2] At first sight, that could seem coherent with the *law of supply and demand*: if the supply increases, the price should go down. . . **assuming the demand is constant** (let's not forget that assumption). That might not be the case, and an increase in both production and demand can result in higher prices. In any case, that so-called law is just an economic model of price determination in a market; as all models, it can fit or not the reality.

```
gasOil$Date <- as.Date(as.character(gasOil$Date), '%Y-%m-%d')
summary(gasOil)
```

```
##       Date              Production        Price      
##  Min.   :1978-01-01   Min.   :119.4   Min.   :1.329  
##  1st Qu.:1986-07-08   1st Qu.:173.0   1st Qu.:1.823  
##  Median :1995-01-16   Median :201.4   Median :2.096  
##  Mean   :1995-01-15   Mean   :210.0   Mean   :2.391  
##  3rd Qu.:2003-07-24   3rd Qu.:255.8   3rd Qu.:2.909  
##  Max.   :2012-02-01   Max.   :283.2   Max.   :4.432  
```

```
round(stat.desc(gasOil[, 2:3], desc = TRUE, norm = TRUE), 2)
```

```
##              Production  Price
## nbr.val          410.00 410.00
## nbr.null           0.00   0.00
## nbr.na             0.00   0.00
## min              119.41   1.33
## max              283.25   4.43
## range            163.84   3.10
## sum            86102.96 980.50
## median           201.44   2.10
## mean             210.01   2.39
## SE.mean            2.07   0.03
## CI.mean.0.95       4.07   0.07
## var             1753.57   0.49
## std.dev           41.88   0.70
## coef.var           0.20   0.29
## skewness           0.17   0.71
## skew.2SE           0.71   2.95
## kurtosis          -1.38  -0.59
## kurt.2SE          -2.87  -1.23
## normtest.W         0.92   0.91
## normtest.p         0.00   0.00
```

```
# Check that all months between start and end dates appear in the dataset
identical(gasOil$Date, seq(min(gasOil$Date), max(gasOil$Date), by='month'))
```

```
## [1] TRUE
```

The dataset contains 410 observations of 3 variables: the first one corresponds to dates (in character format), the second to U.S. oil production (in millions of barrels of oil, ranging from 119.4 to 283.2 millions of barrels), and the third one to inflation-adjusted average gas prices (in U.S. dollars, ranging from 1.33 to 4.43 USD). All dates correspond to the first day of the month (i.e., we have monthly observations of production and price), from January 1978 until February 2012 (i.e., 34 years—from 1978 to 2011—and 2 months—the first 2 moths of 2012). There are no missing values for any month, and all months between the start and end date are available in the dataset.

```
Production <- ts(data = gasOil$Production, start = year(gasOil$Date[1]),
                 frequency = 12)
Price <- ts(data = gasOil$Price, start = year(gasOil$Date[1]), frequency = 12)
```

Oil production was relatively flat (it was actually U-shaped, but it decreased and then increased at a much lower rate than it did afterwards) from 1978 to 1985, then it had a declining trend until 2009 or so, and it has increased (at a lower rate) since then (probably due to the introduction of driling).

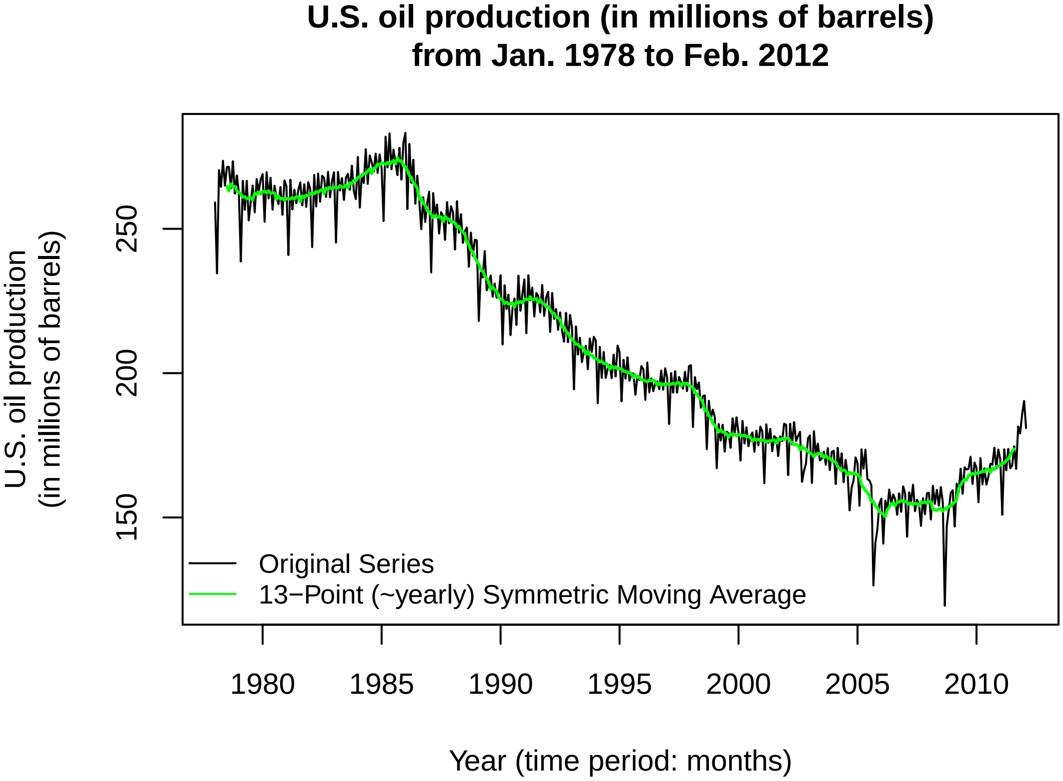


Figure 37: Time series plots of the U.S. oil production (in millions of barrels) from January 1978 to February 2012

As for the inflation-adjusted average gas prices, their dynamics are quite different: they increased a lot (more than $1, almost a 50% increase) from 1978 to 1981 or so, then decreased until 1986-1987 (to levels below the previous ones), remained relatively flat (or even decreased a bit more) unti 1999, and has kept increasing since then, except for a sharp fall at the end of 2008 (that's approximately when the domestic oil production began to increase again; maybe the drop in production was due to the hype about drilling?).

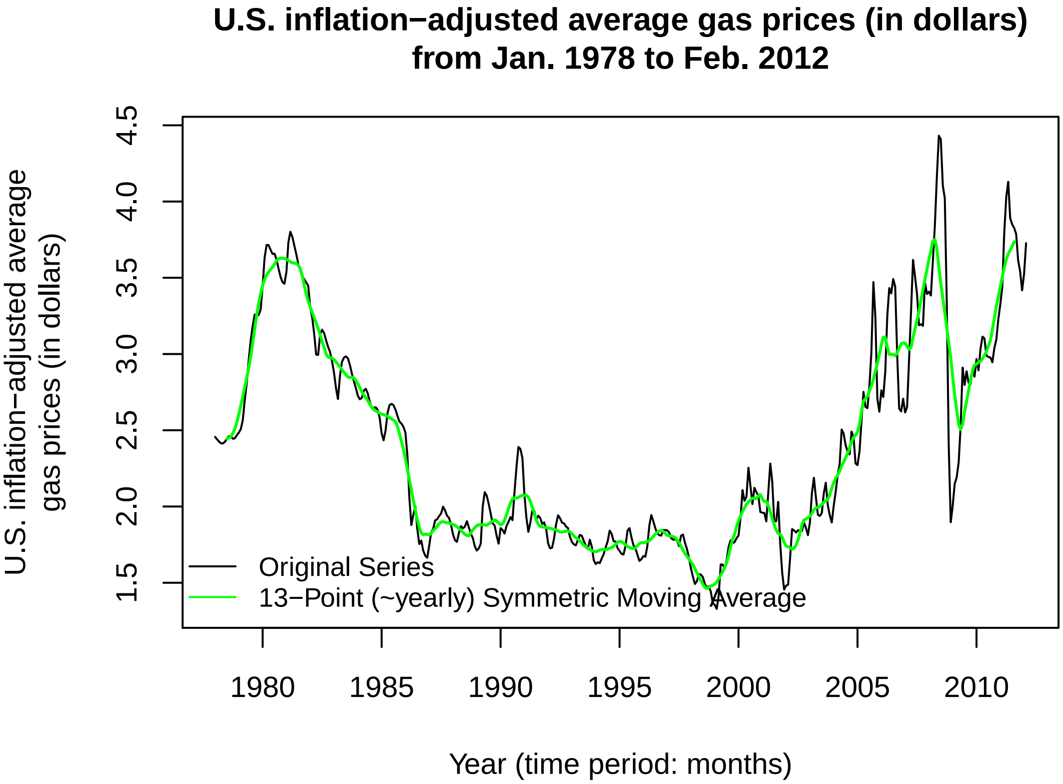


Figure 38: Time series plot of the U.S. inflation-adjusted average gas prices (in dollars) from January 1978 to February 2012

Another thing to notice is that the oil production has much more variability (it fluctuates a lot around its moving average), while the gas price is more persistent. Neither has a clear increasing or decreasing trend but the trend varies over time. Finally, the production seems to have a yearly seasonal component (this is later confirmed when plotting its PACF), while the price does not.

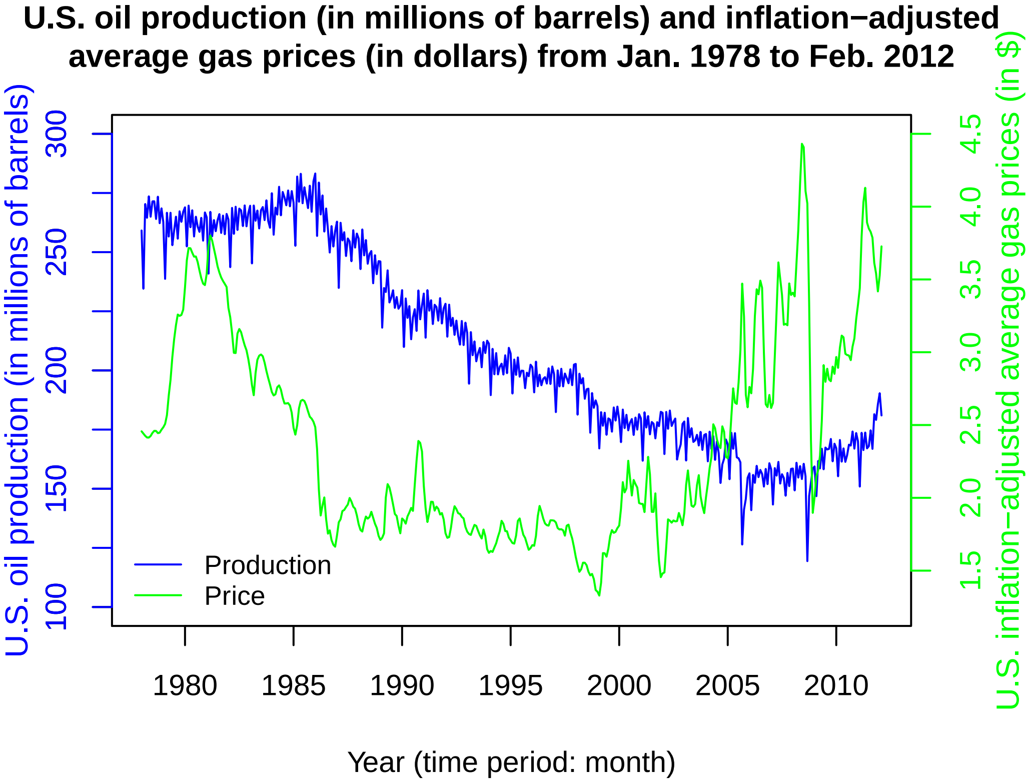


Figure 39: Combined time series plot of the U.S. inflation-adjusted average gas prices (in dollars) and U.S. inflation-adjusted average gas prices (in dollars) from January 1978 to February 2012

The Figure in the next page shows the (approximate) density plots of both time series (one is bimodal and the other is very right-skewed; anyway, density plots tell us nothing about the dynamics of a time series), as well as the correlation (close to zero) and the scatterplot (U-shaped instead of a diagonal line).

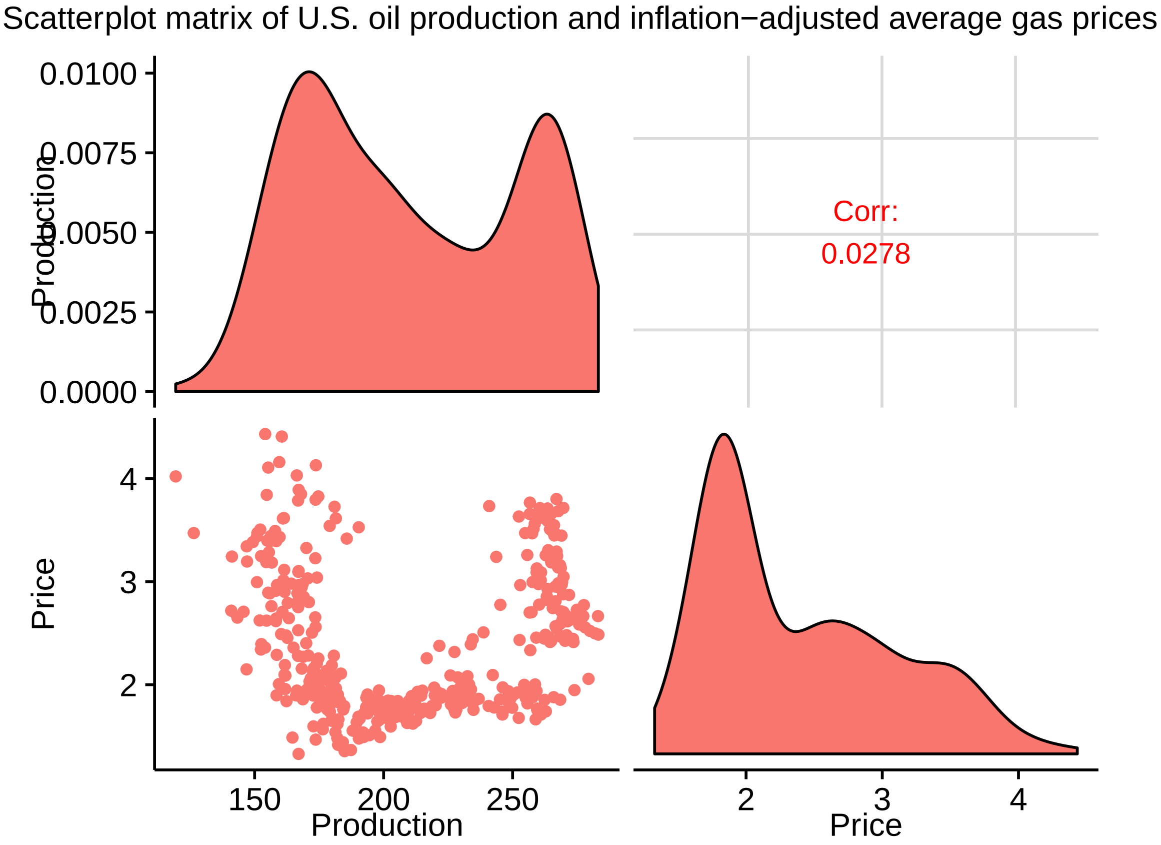


Figure 40: Matrix of the density plots, correlation, and scatterplot of the U.S. oil production and inflation-adjusted average gas prices, from January 1978 to February 2012

Let's now run a Pearson's correlation test to estimate the correlation between both time series, as the AP did, as well the *p*-value and standard error.

```
(ProdPrice.cor <- cor.test(Production, Price))
```

```
##
##  Pearson's product-moment correlation
##
## data:  Production and Price
## t = 0.56088, df = 408, p-value = 0.5752
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.06927648  0.12427029
## sample estimates:
##        cor
## 0.02775705
```

The estimated correlation, as the previous Figure already showed, is about 0.028, not significantly different from zero ($p = 0.575$, and the confidence interval includes zero). That is consistent with the claims from the AP. So the mathematical result, so to speak, is true.

Another way to estimate the correlation is running a linear regression with one of the time series as the regressand and the other as the regressor. The squared root of the R-squared value of the regression is the correlation between both.

```
(linReg <- summary(lm(Price ~ Production)))
```

```
##
## Call:
## lm(formula = Price ~ Production)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0430 -0.5683 -0.2762  0.5287  2.0660
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.2943109  0.1765964  12.992   <2e-16 ***
## Production  0.0004626  0.0008247   0.561    0.575
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6984 on 408 degrees of freedom
## Multiple R-squared:  0.0007705,  Adjusted R-squared:  -0.001679
## F-statistic: 0.3146 on 1 and 408 DF,  p-value: 0.5752
```

```
(ProdPrice.cor2 <- sqrt(linReg$r.squared))
```

```
## [1] 0.02775705
```

What is erroneous (the second error) is the implications from that result (and the use of correlation, to begin with): correlation is not a good measure of the dependency of two time series. Same way that two independent time series can show a high **spurious correlation** (the correlation between both time series is driven by some underlying common driver or it is merely "coincidental"; there are multiple examples), the opposite can happen (though to noise or other factors that may also drive the time series and "mask" their dependence; that's might be the case here: even if U.S. prices depend on domestic production, it's world production—and other possible factors—what drives them).

This goes down to the **definitions of correlation and time series** models. The correlation is defined as the quotient of the covariance of two random variables divided by the product of their respective standard deviations (the square root of their variances). (The sample estimates[3] of) These two parameters—variance and covariance—depend on the value of the individual observations and their (sample) mean, which is constant (and that is not the case in time series!). Now let's revisit what a stochastic process (which is how we model time series) is: it is **a collection of random variables** representing the evolution of some sytem of random values over time. That is (in the case of discrete time series like the ones we are working with), a sequence of random variables, that may be completely different at the different times (and dependent…or not); the only requirement is that those random variables all take values in the same space. To put it simply, it makes no sense to define the mean of $\{x_t\},\ t = 1, \dots, n$[4] because $x_1, x_2, \dots, x_n$ **are not observations from a single random variable**, **but** from a realization of a stochastic process, i.e., **from** n **different random variables, not necessarily i.i.d.**

[3] Which is what we are able to estimate (we are often not able to estimate the population estimates).

[4] The same can be said of the correlation, if we extend this idea to two time series, $\{x_t\}$ and $\{y_t\}$.

What could have been done different? Two time series that are independent and contain unit roots (i.e., they exhibit **stochastic trends**) may show an apparent linear relationship, due to chance similarity of the random walks over the period of the time series. However, it is also possible that those time series are actually related / **cointegrated** (if **a linear combination of them is stationary**). Hence, **we can check if production and price are cointegrated. If we don't have evidence that supports that hypothesis, we also have no evidence of a (linear) relationship. At the same time, the analysis of the inflation-adjusted gas prices will serve us a first step in the creation of a model for the 2nd task.**

As a first step, we plot the ACF and PACF of both time series. They do not suggest that any of the two series is a random walk, since the PACF does not fall sharply after the 1st lag. That would have been the simplest case, but it's always worth exploring it (we could also have plotted the ACF and PACF of the first difference.)

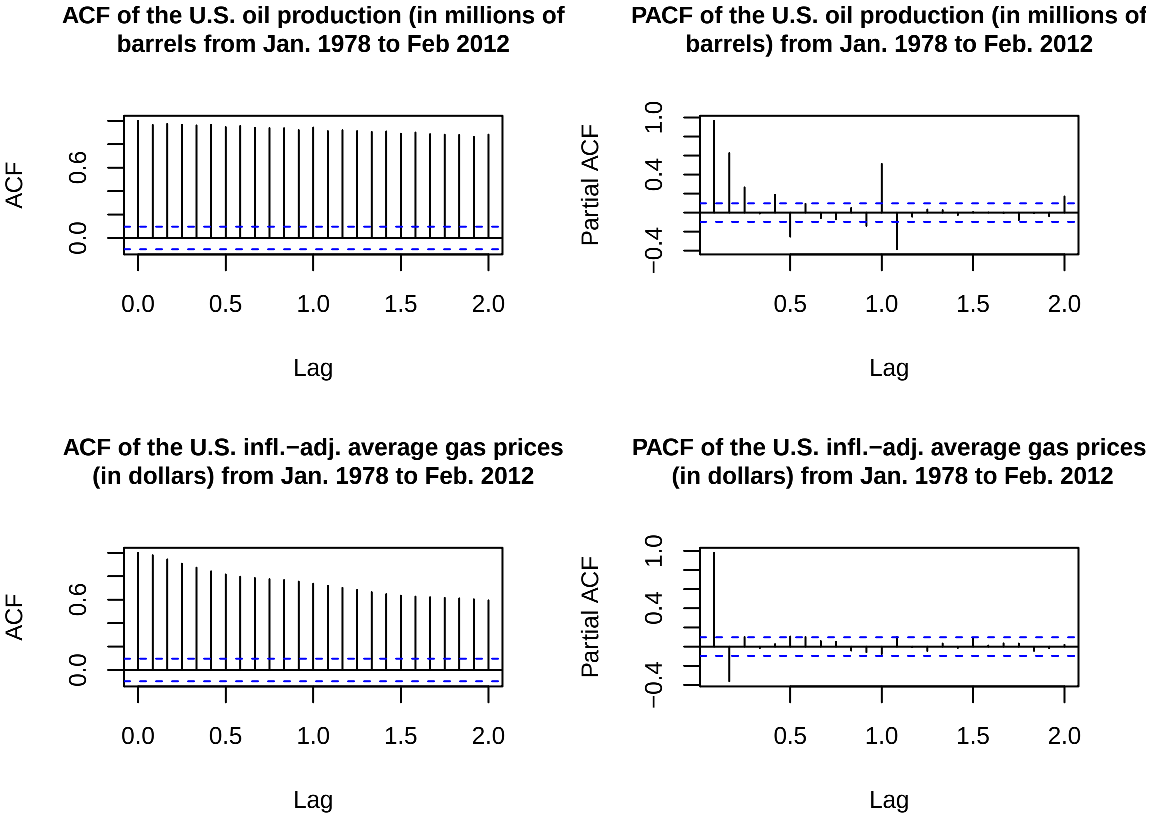


Figure 41: ACF and PACF of the U.S. oil production and inflation-adjusted average gas prices, from January 1978 to February 2012

To test for unit roots, we use the augmented Dickey-Fuller and the Phillips-Perron tests. Based on the p-values of both tests, there is no evidence to reject the unit root hypothesis in the price time series; interestingly, the results of both tests are completely different for the production time series.

```r
# Augmented Dickey-Fuller Test
adf.test(Production)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  Production
## Dickey-Fuller = -0.10686, Lag order = 7, p-value = 0.99
## alternative hypothesis: stationary
```

```
adf.test(Price)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  Price
## Dickey-Fuller = -1.0162, Lag order = 7, p-value = 0.9355
## alternative hypothesis: stationary
```

```
# Phillips-Perron Unit Root Test
pp.test(Production)
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  Production
## Dickey-Fuller Z(alpha) = -124.32, Truncation lag parameter = 5,
## p-value = 0.01
## alternative hypothesis: stationary
```

```
pp.test(Price)
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  Price
## Dickey-Fuller Z(alpha) = -9.5647, Truncation lag parameter = 5,
## p-value = 0.5752
## alternative hypothesis: stationary
```

Now we run a Phillips-Ouliaris test to test the cointegration of both time series. And we find no evidence of cointegration between U.S. oil production and inflation-adjusted average gas prices, from January 1978 to February 2012 (the *p*-value is 0.15 so we cannot reject the null hypothesis that the two series are **not** cointegrated).

```
po.test(gasOil[, 2:3])
```

```
##
##  Phillips-Ouliaris Cointegration Test
##
## data:  gasOil[, 2:3]
## Phillips-Ouliaris demeaned = -3.5509, Truncation lag parameter =
## 4, p-value = 0.15
```

2nd task

Your second task is to create a more statistically-sound model that can be used to predict/forecast inflation-adjusted gas prices. Use your model to forecast the inflation-adjusted gas prices from 2012 to 2016.

In the previous task we already found that the price series is likely to have a unit root, no seasonal component, and possibly an AR component of order 2 (because its PACF falls sharply after that lag).

First we explore ARIMA possible models based on their AIC and BIC values, re-using part of the code we used in **HW 8** with the following changes:

- This time we include integrated series of order d.
- But not SARIMA models since it seems the series has no seasonal component.
- We limit the maximum order of p, d, or q to 3. As we know there is a unit root, the minimum order of d will be 1.

> Same results are found if we include $d = 0$. The same goes for SARIMA models (anyway, including the seasonal components (P, D, and Q) makes this "brute-force" approach take a very long time).

```r
max_coef <- 3
orders <- data.frame(permutations(n = max_coef + 1, r = 3, v = 0:max_coef,
                                  set = FALSE, repeats.allowed = TRUE))
dim(orders)[1] # Number of models up to max_coef
```

```
## [1] 64
```

```r
colnames(orders) <- c("p", "d", "q")
orders <- orders %>% dplyr::filter(d >= 1)
dim(orders)[1] # Number of models considered
```

```
## [1] 48
```

```r
orders %>% sample_n(10) # A 10-sample of the possible orders
```

```
##    p d q
## 6  0 2 1
## 30 2 2 1
## 29 2 2 0
## 46 3 3 1
## 38 3 1 1
## 28 2 1 3
## 1  0 1 0
## 10 0 3 1
## 27 2 1 2
## 21 1 3 0
```

```r
model_list <- orders %>% rowwise() %>%
  mutate(aic = try_default(AIC(Arima(Price, order = c(p, d, q))), default = NA,
                           quiet = TRUE))
model_list <- model_list %>% dplyr::filter(!is.na(aic))
dim(model_list)[1] # Number of models estimated
```

```
## [1] 47
```

```
model_list <- model_list %>%
  mutate(bic = BIC(Arima(Price, order = c(p, d, q))))
```

Table 10: Top 5 models based on the (lowest) AIC value

p	d	q	aic	bic
1	1	3	-675.6	-655.6
2	1	3	-675.2	-651.1
3	1	3	-674.1	-646.0
1	1	2	-670.8	-654.7
0	1	3	-670.4	-654.3

Table 11: Top 5 models based on the (lowest) BIC value

p	d	q	aic	bic
1	1	3	-675.6	-655.6
0	1	2	-667.4	-655.4
2	1	0	-667.3	-655.2
1	1	2	-670.8	-654.7
0	1	3	-670.4	-654.3

The ARIMA model with the lowest AIC and BIC values is ARIMA(1,1,3). The 2nd and 3rd best models, based on their AIC value (very close to the former), are ARIMA(2,1,3) and ARIMA(3,1,3); the increased number of parameters is penalized by the BIC, so those do not appear in the Top 5 models based on the BIC value: that criterion selects ARIMA(0,1,2) and ARIMA(2,1,0) as the 2nd and 3rd best models, respectively. These 5 models will be our potential candidates.

```
orders_AIC <- model_list %>% arrange(aic) %>% top_n(-3, aic) %>% select(p, d, q)
orders_BIC <- model_list %>% arrange(bic) %>% top_n(-3, bic) %>% select(p, d, q)
orders <- rbind_list(orders_AIC, orders_BIC) %>% unique()
models <- apply(orders, 1, function(arima_order)
  Arima(Price, order = c(arima_order[1], arima_order[2], arima_order[3])))
```

If we use the `auto.arima()` function instead of our own loop the best model based on the AIC value is still the ARIMA(1,1,3)… even if we include the seasonal components (so our decision of excluding them seems correct). The best model based on the BIC value is the ARIMA(0,1,2) (possibly because the function uses other method by default different from `Arima()`).

```
auto.arima(Price, seasonal = TRUE, ic = "aic") # same result using AICc
```

```
## Series: Price
## ARIMA(1,1,3)
##
## Coefficients:
##          ar1      ma1      ma2      ma3
##       0.7578  -0.1455  -0.3948  -0.2355
## s.e.  0.0947   0.1039   0.0565   0.0508
```

```
##
## sigma^2 estimated as 0.01104:  log likelihood=342.82
## AIC=-675.65   AICc=-675.5   BIC=-655.58
```

```
auto.arima(Price, seasonal = TRUE, ic = "bic")
```

```
## Series: Price
## ARIMA(0,1,2)
##
## Coefficients:
##          ma1     ma2
##       0.6453  0.1767
## s.e.  0.0531  0.0516
##
## sigma^2 estimated as 0.01133:  log likelihood=336.7
## AIC=-667.41   AICc=-667.35   BIC=-655.37
```

As we know, the lowest AIC or BIC value may not necessarily involve the best model (with highest explanatory power), especially when we're interested in forecasting. That criterion should be combined with others, such as how much the residuals of the model resemble a white noise (and then selecting the simplest model among those). So next we examine the ACFs and PACFs of the residuals of all these models.

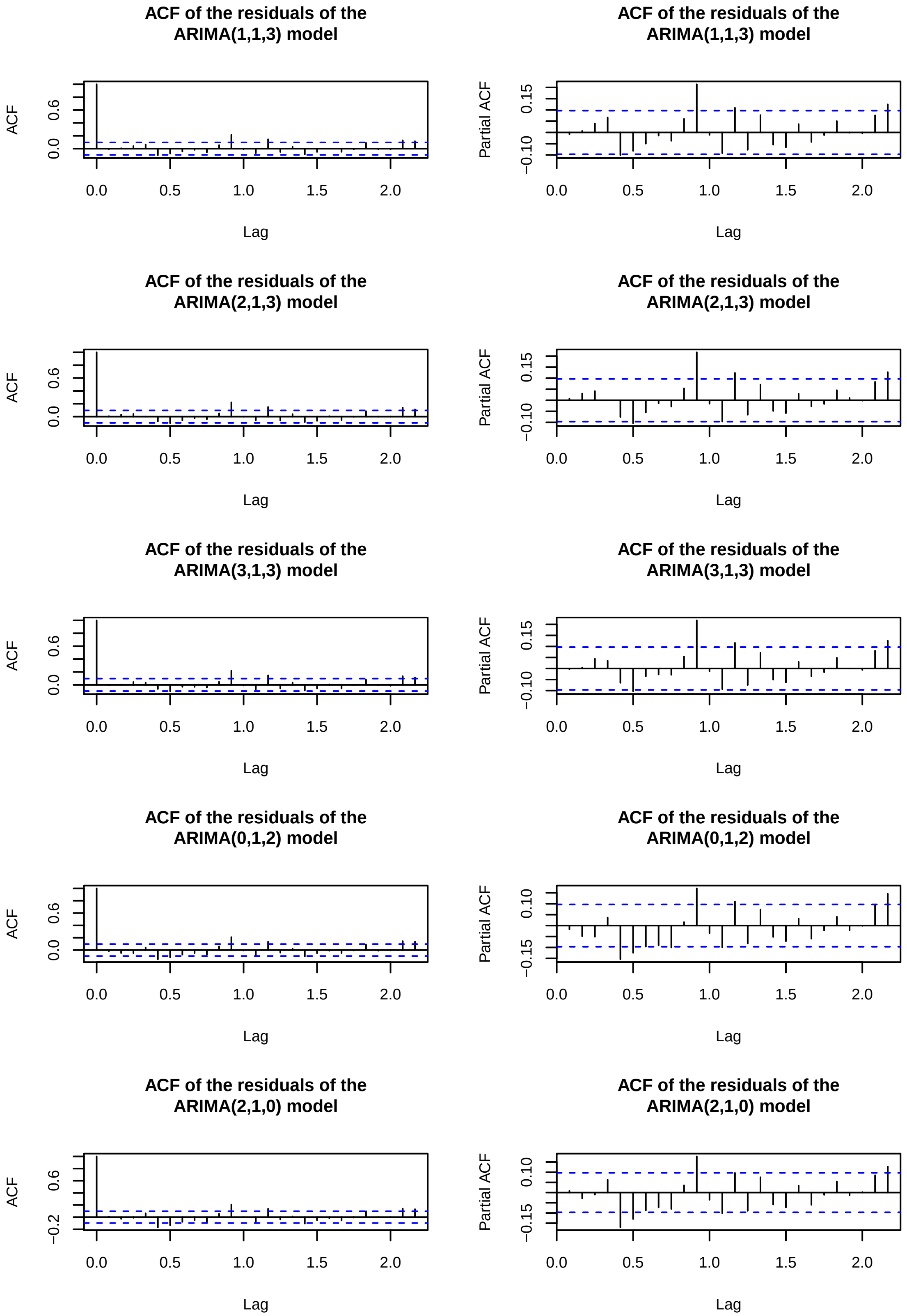


Figure 42: ACF and PACF of the 5 candidates models for the U.S. inflation-adjusted average gas prices, from January 1978 to February 2012

If the residuals were a white noise, only 5% of the auto-correlations (or partial auto-correlations), on average, would be significant (by mere chance). That would correspond to 1 (or 2 at the most) significant auto-correlations (or partial auto-correlations) at the 24 lags we have plotted, but the previous Figure show that 3 or 4 (up to 5 or 6 for the last 2 models, ARIMA(0,1,2) and ARIMA(2,1,0)) are significant. **Anyway, the significant auto-correlations** (which, for the best 3 models based on the AIC value, always occur from lag 9 on) **have a relatively low value (0.2 or so)**, so we'll assume the residuals of those 3 models approximately ensemble a white noise.

As it happened in **HW8**, most of the plots do not differ too much between one another, so selecting the best is difficult after a visual inspection. We complement that with our previous approach: explore the sum of the absolute value of all the auto-correlations (and partial auto-correlations) that are significant (i.e., that exceed $2/\sqrt{n}$, the variance of the lag k autocorrelation—ρ_k—of a white noise, in absolute value).

```r
sum_acf <- function(model) {
  # Get the ACFs of first 24 lags
  ACF <- stats::acf(model$residuals, plot = FALSE, lag.max = 24)$acf
  # Exclude (assign 0) to those not significant
  significant_ACF <- ifelse(abs(ACF) < qnorm(.975) / sqrt(model$nobs), 0,
                            abs(ACF))
  # Sum absolute values (exluding lag 0)
  return(sum(significant_ACF[-1]))
}
sum_pacf <- function(model) {
  # Get the PACFs of first 24 lags
  PACF <- pacf(model$residuals, plot = FALSE, lag.max = 24)$acf
  # Exclude (assign 0) to those not significant
  significant_PACF <- ifelse(abs(PACF) < qnorm(.975) / sqrt(model$nobs), 0,
                             abs(PACF))
  # Sum absolute values
  return(sum(significant_PACF))
}
model_list <- join(orders, model_list, by=c("p","d","q"), type="inner") %>%
  rowwise() %>%
  mutate(ACF = sum_acf(Arima(Price, order = c(p, d, q))),
         PACF = sum_pacf(Arima(Price, order = c(p, d, q))))
```

Table 12: Top 5 models based on the (lowest) sum of the absolute value of their (significant) auto-correlations

p	d	q	aic	bic	ACF	PACF
1	1	3	-675.6	-655.6	0.5	0.4
3	1	3	-674.1	-646.0	0.5	0.4
2	1	3	-675.2	-651.1	0.5	0.4
0	1	2	-667.4	-655.4	0.7	0.8
2	1	0	-667.3	-655.2	0.8	0.6

The Table above confirms our visual inspection of the plots in the Figure of the previous page: the 3 models with the lowest AIC value are similar in terms of the ACF and PACF of their residuals, and the other 2 models (2nd and 3rd best models based on the BIC value) are worse in terms of their residuals.

So the ARIMA(0,1,2) and ARIMA(2,1,0) models have similar BIC values than the ARIMA(1,1,3) model, and they are less complex (in terms of the number of coefficients (2 vs. 4), but their AIC values are higher (and hence worse; though this is not a critical issue; we are interested in the best predictions of future values, not the best fitting of the past ones) and (more importantly) their residuals do not resemble white noise so well. As for the ARIMA(2,1,3) and ARIMA(3,1,3) models, their AIC values are almost equal to that of the ARIMA(1,1,3) model and their residuals look almost the same, but their BIC values are much higher and they are more complex (5 and 6 coefficients vs. 4). Summarizing, we select the ARIMA(1,1,3) model as the best candidate; we'll also try the (much simpler) ARIMA(0,1,2) model.

To select between these 2 models we will also analyze their out-of-sample fit (we'll omit the in-sample fit for the whole time period; the results for the training set used in the out-of-sample fit look pretty similar). To train the models we will use approximately 90% of the original observations, 31 years, leaving out the last 38 months.

```
models <- models[c(1,4)]
orders <- orders[c(1,4), ]
Price.train <- window(Price, start = 1978, end=c(2008, 12))
Price.test <- window(Price, start = 2009)
(arima113.oos.fit <- Arima(Price.train, order = as.numeric(orders[1, ])))
```

```
## Series: Price.train
## ARIMA(1,1,3)
##
## Coefficients:
##          ar1     ma1     ma2     ma3
##      -0.7445  1.4761  0.6657  0.0159
## s.e.   0.1290  0.1387  0.1446  0.0737
##
## sigma^2 estimated as 0.009951:  log likelihood=330.35
## AIC=-650.71   AICc=-650.54   BIC=-631.13
```

Time	Original series	Estimated series	Residuals
Jan 1978	2.46	2.45	0.00
Feb 1978	2.44	2.45	-0.01
Mar 1978	2.43	2.43	-0.01
Apr 1978	2.41	2.42	-0.01
May 1978	2.41	2.41	0.00
Jun 1978	2.42	2.42	0.01

```
arima113.oos.fit.fcast <- forecast.Arima(arima113.oos.fit, h = 38)
```

Table 14: Goodness-of-fit parameters for the training and test sets (ARIMA(1,1,3)

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	-0.0009231	0.0990829	0.0620179	-0.0565951	2.659808	0.226098	0.0017451
Test set	1.3127492	1.4127668	1.3127492	40.0112148	40.011215	4.785874	0.8837268

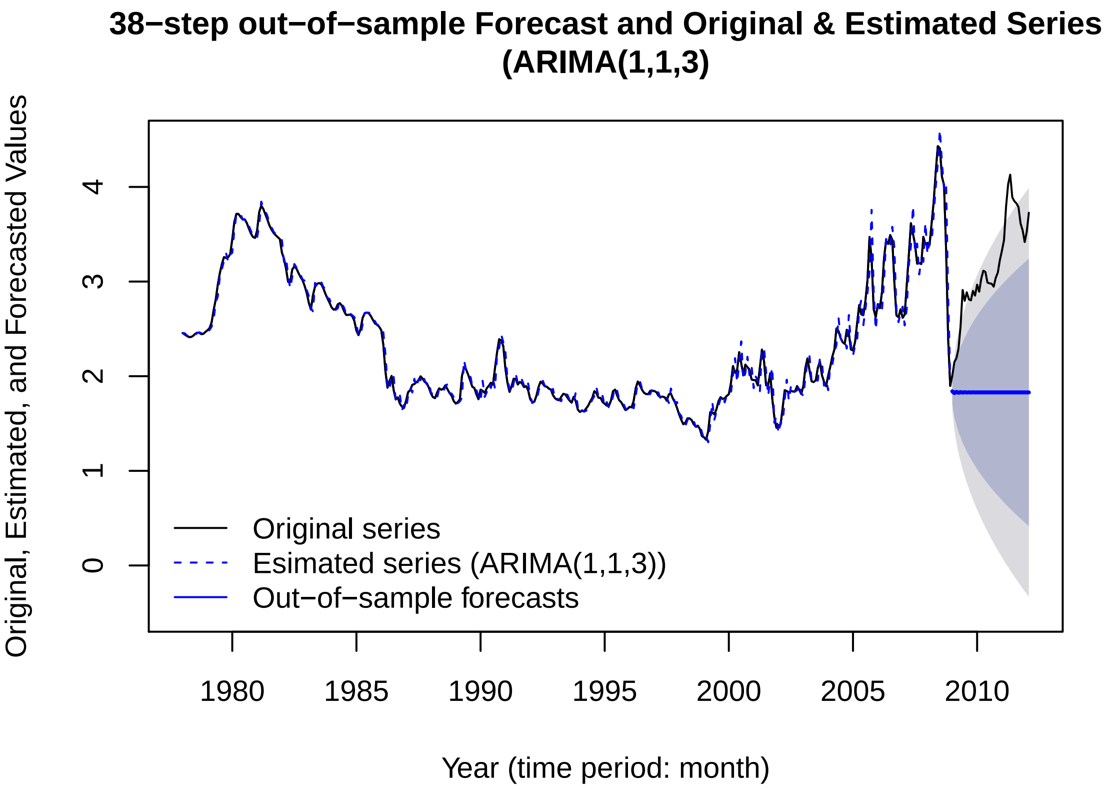


Figure 43: Out-of-sample fit of the ARIMA(1,1,3) model to the U.S. inflation-adjusted average gas prices (in dollars)

```
(arima012.oos.fit <- Arima(Price.train, order = as.numeric(orders[2, ])))
```

```
## Series: Price.train
## ARIMA(0,1,2)
##
## Coefficients:
##          ma1     ma2
##       0.7215  0.1895
## s.e.  0.0568  0.0511
##
## sigma^2 estimated as 0.0103:  log likelihood=323.15
## AIC=-640.3   AICc=-640.23   BIC=-628.55
```

Time	Original series	Estimated series	Residuals
Jan 1978	2.46	2.45	0.00
Feb 1978	2.44	2.45	-0.01
Mar 1978	2.43	2.43	-0.01
Apr 1978	2.41	2.42	-0.00
May 1978	2.41	2.41	0.00
Jun 1978	2.42	2.42	0.01

```
arima012.oos.fit.fcast <- forecast.Arima(arima012.oos.fit, h = 38)
```

Table 16: Goodness-of-fit parameters for the training and test sets (ARIMA(0,1,2)

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	-0.0009427	0.1010619	0.0625097	-0.0504004	2.655063	0.2278907	-0.0192977
Test set	1.3622718	1.4588620	1.3622718	41.6361560	41.636156	4.9664178	0.8845666

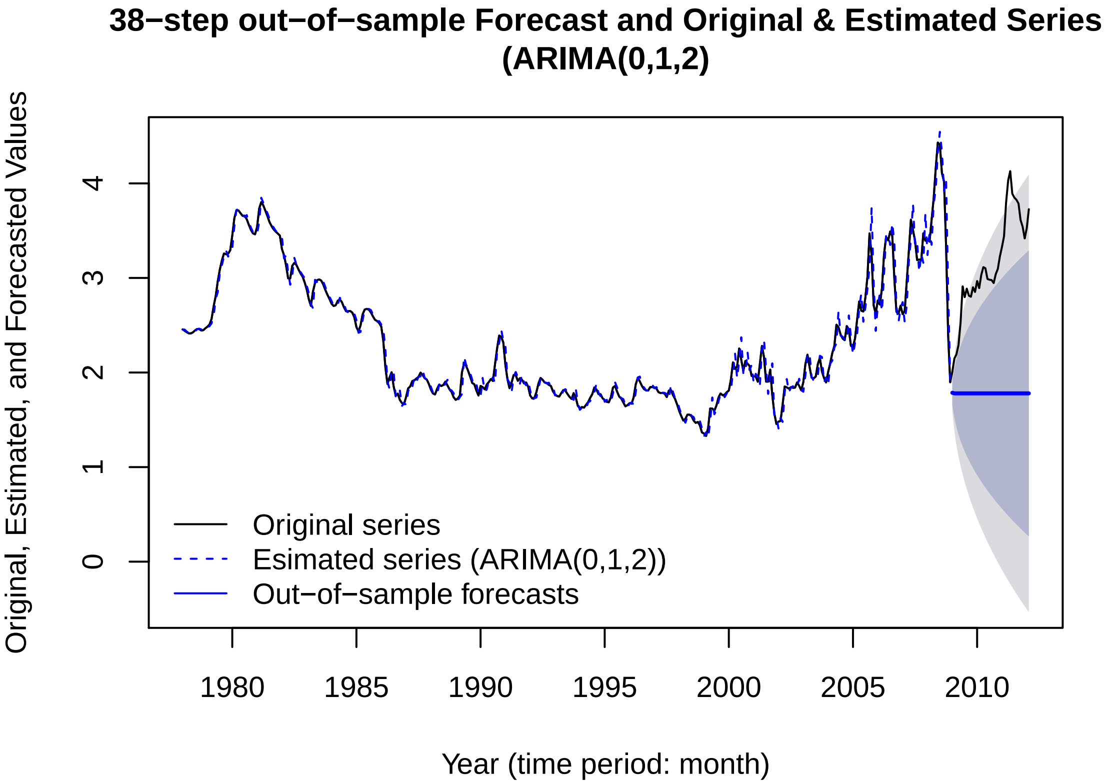


Figure 44: Out-of-sample fit of the ARIMA(0,1,2) model to the U.S. inflation-adjusted average gas prices (in dollars)

The last 2 Figures look very similar. For both models:

- the in-sample fit (of the training set) is very good,
- the mean value of the forecasts is (almost) constant, equal to the last value in the training set, and
- the "real" values in the test set fall within the confidence region of the forecasts, except for the peak in the middle of 2011.

We have to **look at the RMSE, MAE, and other goodness-of-fit parameters in the Tables previously shown** to see that the ARIMA(1,1,3) is a better fit (though the differences with the ARIMA(0,1,2) model are small). Hence, that's the model we'll use to forecast the inflation-adjusted gas prices from 2012 to 2016.

```
(arima113.fit <- models[[1]])
```

```
## Series: Price
## ARIMA(1,1,3)
##
## Coefficients:
##          ar1      ma1      ma2      ma3
##       0.7578  -0.1455  -0.3948  -0.2355
## s.e.  0.0947   0.1039   0.0565   0.0508
##
## sigma^2 estimated as 0.01104:  log likelihood=342.82
## AIC=-675.65   AICc=-675.5   BIC=-655.58
```

Table 17: Coefficients, SEs, and 95% CIs of the estimated ARIMA(1,1,3) model

	Coefficient	SE	95% CI lower	95% CI upper
ar1	0.7578	0.0947	0.5685	0.9472
ma1	-0.1455	0.1039	-0.3533	0.0623
ma2	-0.3948	0.0565	-0.5078	-0.2818
ma3	-0.2355	0.0508	-0.3371	-0.1339

```
arima113.fit.fcast <- forecast.Arima(arima113.fit, h = 58)
pander(predict(arima113.fit, n.ahead = 58)$pred)
```

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2012	NA	NA	3.823	3.815	3.784	3.761	3.744	3.731	3.721	3.713	3.708	3.703
2013	3.7	3.697	3.696	3.694	3.693	3.692	3.692	3.691	3.691	3.69	3.69	3.69
2014	3.69	3.69	3.69	3.69	3.69	3.69	3.69	3.69	3.69	3.69	3.69	3.69
2015	3.69	3.69	3.69	3.69	3.69	3.69	3.69	3.69	3.69	3.69	3.69	3.69
2016	3.69	3.69	3.69	3.69	3.69	3.69	3.69	3.69	3.69	3.69	3.69	3.69

I.e, our model, ARIMA(1,1,3), for $\{x_t\}$ (where x_t is the U.S. inflation-adjusted average gas prices (in dollars) at time t) is:

$$\Theta_1(B)(1-B)^1 x_t = \Phi_3(B)\omega_t$$

$$(1 - B - 0.758B)(1 - B)x_t = (1 + -0.145B + -0.395B^2 + -0.236B^3)\omega_t$$

$$x_t = 1.758x_{t-1} - 0.758x_{t-2} + \omega_t - 0.145\omega_{t-1} - 0.395\omega_{t-2} - 0.236\omega_{t-3}$$

where $\{w_t\}$ is a white noise series with mean zero and variance σ^2.

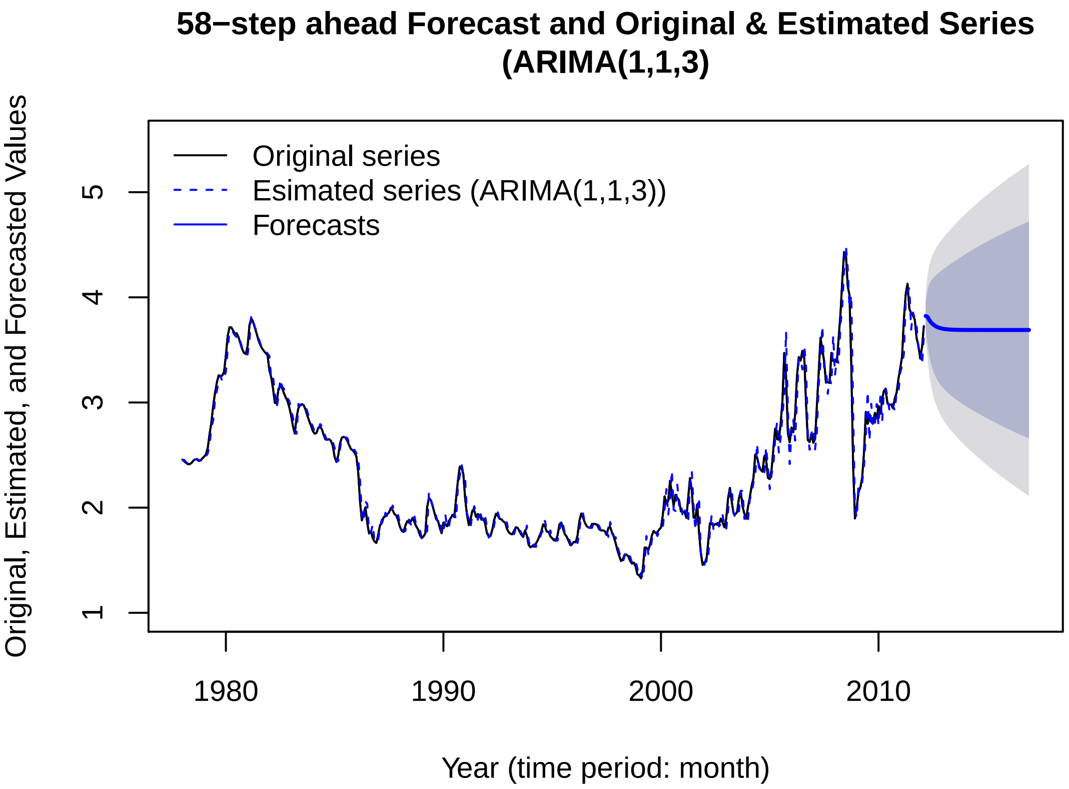


Figure 45: 58-step ahead forecasts (from March 2012 to December 2016) of the U.S. inflation-adjusted average gas prices (in dollars) based on an ARIMA(1,1,3) model fitted to data from January 1978 to February 2012

But we haven't checked for conditional heteroskedasticity (volatility) yet, so the prediction intervals above might not be accurate. After checking the auto-correlations of the squared residuals of our model (see the 1st Figure in the following page) we find that many of them are significant, which indicates volatility, so we start applying a GARCH(1,1) model.

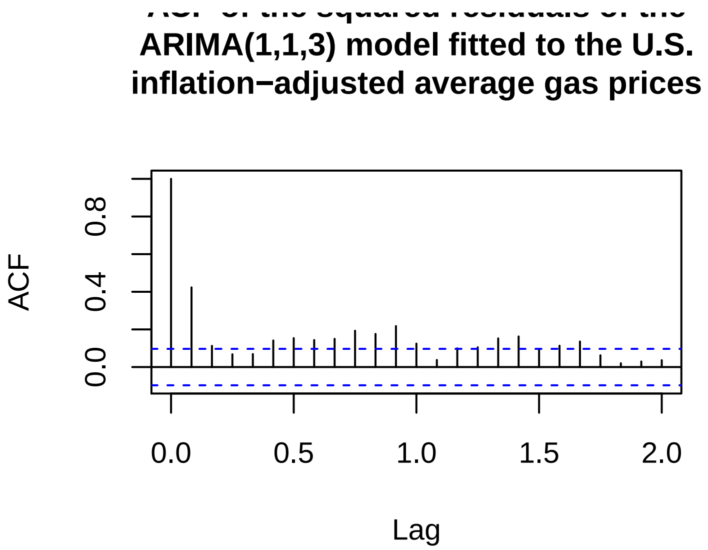


Figure 46: ACF of the squared residuals of the ARIMA(1,1,3) model fitted to the U.S. inflation-adjusted average gas prices

```
(Price.garch11 <- garch(resid(arima113.fit), trace = FALSE))
```

```
##
## Call:
## garch(x = resid(arima113.fit), trace = FALSE)
##
## Coefficient(s):
##        a0         a1         b1
## 0.0002877  0.2158131  0.7753565
```

```
Price.garch11.res <- Price.garch11$res[-1]
t(confint(Price.garch11))
```

```
##                  a0        a1        b1
## 2.5 %  0.0001442508 0.1219635 0.6961534
## 97.5 % 0.0004312333 0.3096627 0.8545597
```

As shown below, the residuals of that GARCH(1,1) model fairly resemble a white noise (the auto-correlation at lag 16 is significant, but about 5% of them could be, just due to chance), so we can use this model GARCH(1,1).

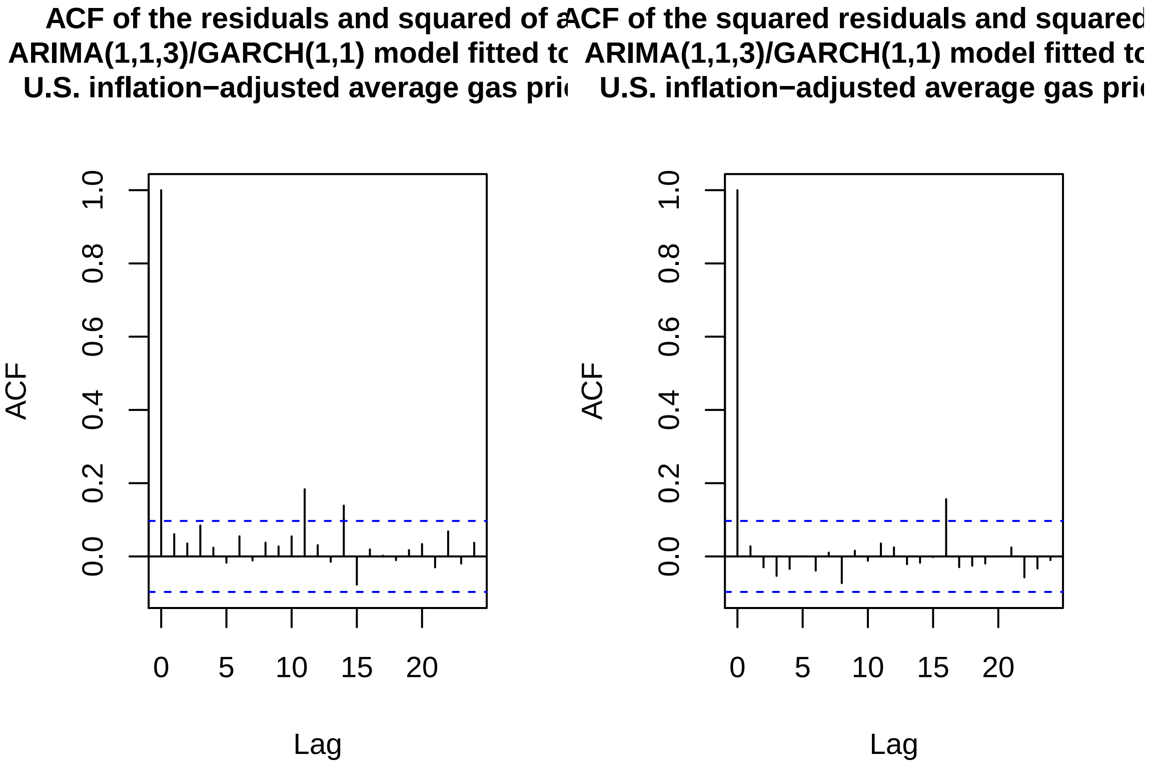


Figure 47: ACF of the residuals and squared residuals of an ARIMA(1,1,3)/GARCH(1,1) model fitted to the U.S. inflation-adjusted average gas prices

Hence, our complete model, ARIMA(1,1,3)/GARCH(1,1) is:

$$x_t = 1.758x_{t-1} - 0.758x_{t-2} + \epsilon_t - 0.145\epsilon_{t-1} - 0.395\epsilon_{t-2} - 0.236\epsilon_{t-3}$$

where

$$\epsilon_t = \omega_t \sqrt{h_t}$$

and

$$h_t = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \beta_1 h_{t-1} = 0.000288 + 0.215813\epsilon_{t-1}^2 + 0.775357h_{t-1}$$

($\{\omega_t\}$ is again a white noise wiht zero mean, but now with unit variance; the variance of the error term—now called ϵ_t—at each moment is h_t.)

Next we estimate the conditional variance of the series ($h_t = \sigma_t^2$), and confirm how it changes with time (especially after 2000).

```
ht <- Price.garch11$fit[,1]^2 # conditional variance
```

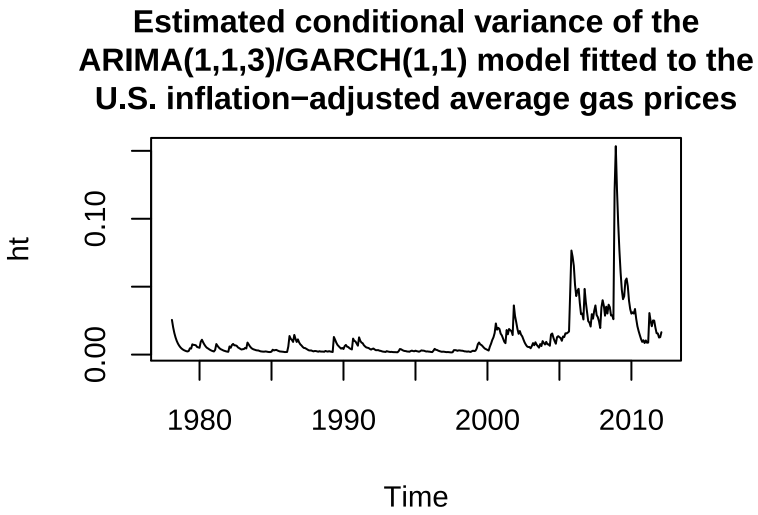


Figure 48: Estimated conditional variance of the ARIMA(1,1,3)/GARCH(1,1) model fitted to the U.S. inflation-adjusted average gas prices

Finally, we have to predict the variance for the 58 months until the end of 2016. **For a complete explanation about how to re-adjust the prediction intervals, see Part 2; here, what we add here is our own code (before using fGarch)**.

```
res.CI.halfwidth <- qnorm(.975) * sqrt(ht) # CI of epsilon_t
# Variation of Price during observation period
Price.lower <- fitted.values(arima113.fit) - res.CI.halfwidth
Price.upper <- fitted.values(arima113.fit) + res.CI.halfwidth
# Forecasts
# Initialize h_t (cond. variance) and epsilon_t (residuals or error term)
# 58 elements (as many as forecasts)
ht.fcst <- res.fcst <- rep(0, 58)
for (i in 1:58) {
  if (i == 1) { # use last observation
    ht.fcst[i] <- Price.garch11$coef[1] +
      Price.garch11$coef[2] * resid(arima113.fit)[length(Price)]^2 +
      Price.garch11$coef[3] * ht[length(Price)]
  } else { # use previous predictions
    ht.fcst[i] <- Price.garch11$coef[1] +
      Price.garch11$coef[2] * res.fcst[i-1]^2 +
      Price.garch11$coef[3] * ht.fcst[i-1]
  }
  res.fcst[i] <- sqrt(ht.fcst[i]) # epsilon_t = omega_t * sqrt(h_t)
}
# Compare the previous std. dev. with the (changing) new one
sd(resid(arima113.fit))
```

```
## [1] 0.1045261
```

```
c(head(sqrt(ht.fcst)), tail(sqrt(ht.fcst)))
```

```
##  [1] 0.1239134 0.1245258 0.1251299 0.1257258 0.1263137 0.1268936 0.1473809
##  [8] 0.1477060 0.1480276 0.1483456 0.1486601 0.1489712
```

```
# Lower & upper limits of the Price forecasts CI
Price.fcst.lower <- as.numeric(arima113.fit.fcast$mean) -
  c(arima113.fit.fcast$upper[, '95%'] - arima113.fit.fcast$mean) /
  sd(resid(arima113.fit.fcast)) * sqrt(ht.fcst)
Price.fcst.upper <- as.numeric(arima113.fit.fcast$mean) +
  c(arima113.fit.fcast$upper[, '95%'] - arima113.fit.fcast$mean) /
  sd(resid(arima113.fit.fcast)) * sqrt(ht.fcst)
```

And now we can plot the original series and the forecasts until 2016, with the 95% confidence intervals for both periods.

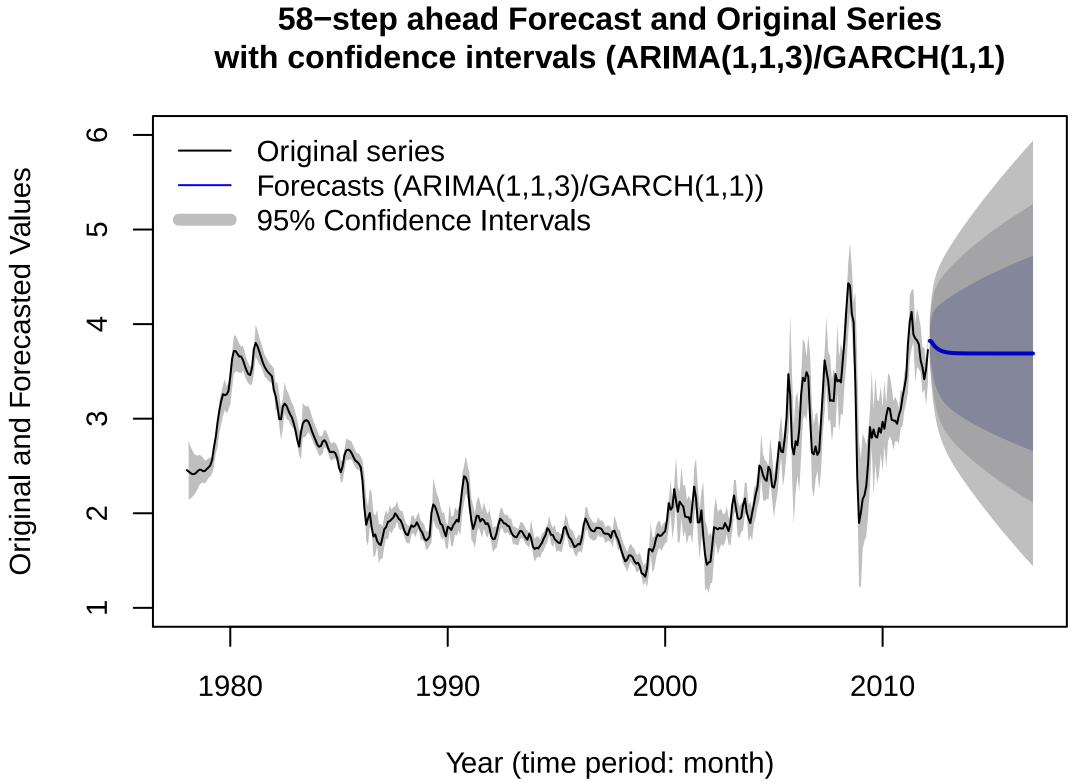


Figure 49: 58-step ahead forecasts (from March 2012 to December 2016) of the U.S. inflation-adjusted average gas prices (in dollars) based on an ARIMA(1,1,3)/GARCH(1,1) model fitted to data from January 1978 to February 2012. Widest gray area corresponds to the 95% confidence region using GARCH, which overlaps the previous 95% (and 80%) regions using only ARIMA

The figure below shows the previous 80% and 95% confidence intervals using ARIMA only: the widest (and lightest) area, that overlaps the other two, corresponds to the new 95% prediction region, much wider.

The `fGarch` package also allows to make predictions of GARCH models, so we start using it on the residuals of our original ARIMA(1,1,3) model:

```
(Price.garch11.2 <- garchFit(~ garch(1,1), data = resid(arima113.fit),
                             trace = FALSE))
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = resid(arima113.fit),
##     trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x10241e20>
##  [data = resid(arima113.fit)]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##          mu        omega       alpha1        beta1
## -0.00167114   0.00029377   0.21332844   0.77490094
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##          Estimate  Std. Error  t value Pr(>|t|)
## mu     -0.0016711   0.0032399   -0.516  0.60599
## omega   0.0002938   0.0001272    2.310  0.02090 *
## alpha1  0.2133284   0.0753299    2.832  0.00463 **
## beta1   0.7749009   0.0699617   11.076  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  454.3815    normalized:  1.108247
##
## Description:
##  Fri Apr 22 17:14:58 2016 by user:
```

```
# res.fcst2 <- predict(Price.garch11.2, n.ahead=58, plot = TRUE, conf = .95)
res.fcst.2 <- predict(Price.garch11.2, n.ahead=58, conf = .95)
Price.fcst.lower.2 <- arima113.fit.fcast$mean + res.fcst.2$meanForecast -
  c(arima113.fit.fcast$upper[, '95%'] - arima113.fit.fcast$mean) /
  sd(resid(arima113.fit.fcast)) * res.fcst.2$standardDeviation
Price.fcst.upper.2 <- arima113.fit.fcast$mean + res.fcst.2$meanForecast +
  c(arima113.fit.fcast$upper[, '95%'] - arima113.fit.fcast$mean) /
  sd(resid(arima113.fit.fcast)) * res.fcst.2$standardDeviation
```

The 95% confidence intervals of the forecasts are quite similar (i.e., we've been able to replicate the predictions of fGarch with our own code :).

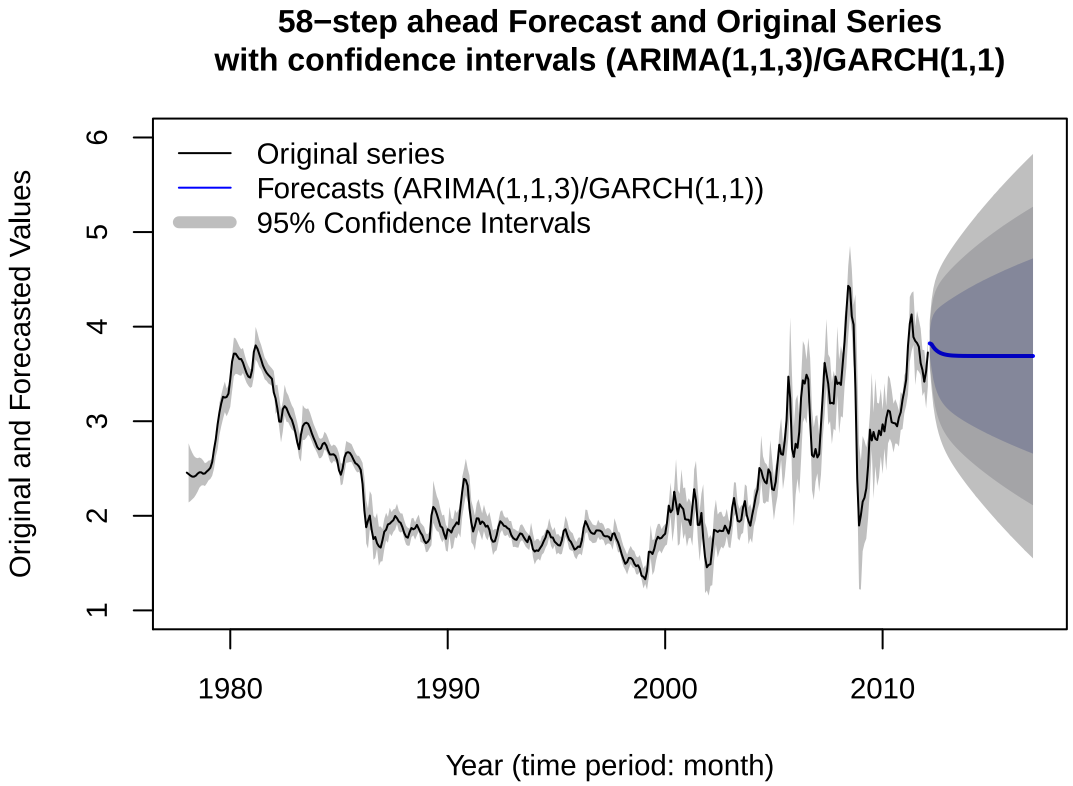


Figure 50: (2nd version of) 58-step ahead forecasts (from March 2012 to December 2016) of the U.S. inflation-adjusted average gas prices (in dollars) based on an ARIMA(1,1,3)/GARCH(1,1) model fitted to data from January 1978 to February 2012. Widest gray area corresponds to the 95% confidence region using GARCH, which overlaps the previous 95% (and 80%) regions using only ARIMA

We can also apply an ARMA(1,3)/GARCH(1,1) on the original series differentiated (`fGarch` only allows to combine ARMA and GARCH models, not ARIMA).

```
(Price.garch11.3 <- garchFit(~ arma(1,3) + garch(1,1), data = diff(Price),
                              trace = FALSE))
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(1, 3) + garch(1, 1), data = diff(Price),
##     trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ arma(1, 3) + garch(1, 1)
## <environment: 0xe7f9e50>
##  [data = diff(Price)]
##
## Conditional Distribution:
```

```
##  norm
##
## Coefficient(s):
##          mu          ar1          ma1          ma2          ma3
## -0.00011153   0.71547907  -0.05543106  -0.38365694  -0.18357647
##       omega       alpha1        beta1
##  0.00023501   0.17377016   0.81366803
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##          Estimate  Std. Error  t value Pr(>|t|)
## mu     -0.0001115   0.0012419   -0.090 0.928442
## ar1     0.7154791   0.1404980    5.092 3.53e-07 ***
## ma1    -0.0554311   0.1470199   -0.377 0.706151
## ma2    -0.3836569   0.0905220   -4.238 2.25e-05 ***
## ma3    -0.1835765   0.0541993   -3.387 0.000706 ***
## omega   0.0002350   0.0001119    2.101 0.035662 *
## alpha1  0.1737702   0.0640187    2.714 0.006640 **
## beta1   0.8136680   0.0625801   13.002  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  454.048    normalized:  1.110142
##
## Description:
##  Fri Apr 22 17:14:58 2016 by user:
```

```r
res.fcst.3 <- predict(Price.garch11.3, n.ahead=58, conf = .95)
# Add the mean prediction of GARCH (close to zero) to the prediction of SARIMA
# and subtract/add the previous CI / sigma * sigma_t
Price.fcst.lower.3 <- arima113.fit.fcast$mean + res.fcst.3$meanForecast -
  c(arima113.fit.fcast$upper[, '95%'] - arima113.fit.fcast$mean) /
  sd(resid(arima113.fit.fcast)) * res.fcst.3$standardDeviation
Price.fcst.upper.3 <- arima113.fit.fcast$mean + res.fcst.3$meanForecast +
  c(arima113.fit.fcast$upper[, '95%'] - arima113.fit.fcast$mean) /
  sd(resid(arima113.fit.fcast)) * res.fcst.3$standardDeviation
```

As expected, the results are quite similar than in the 2 previous cases.

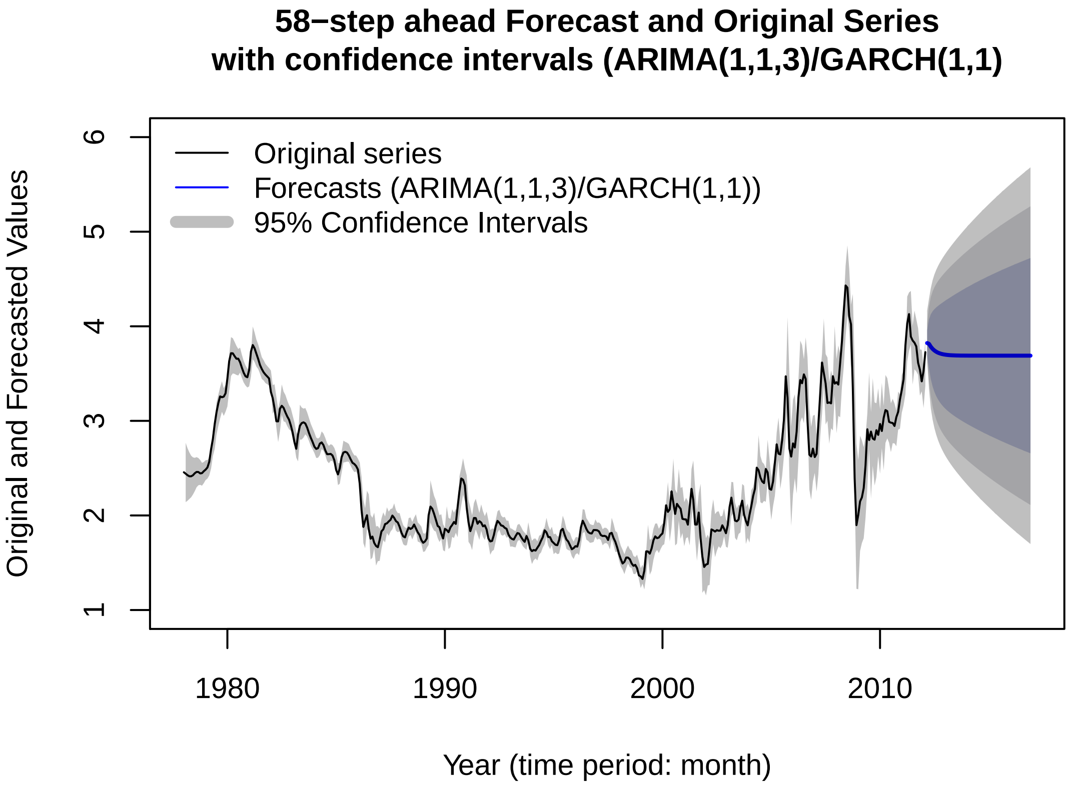


Figure 51: (3rd version of) 58-step ahead forecasts (from March 2012 to December 2016) of the U.S. inflation-adjusted average gas prices (in dollars) based on an ARIMA(1,1,3)/GARCH(1,1) model fitted to data from January 1978 to February 2012. Widest gray area corresponds to the 95% confidence region using GARCH, which overlaps the previous 95% (and 80%) regions using only ARIMA

For comparison, let's just finish plotting the standard deviation of the forecasts for the 3 approaches (our own and two using the `fGarch` library):

```
head(cbind(sqrt(ht.fcst), res.fcst.2$standardDeviation,
          res.fcst.3$standardDeviation))
```

```
##             [,1]      [,2]      [,3]
## [1,] 0.1239134 0.1237363 0.1264679
## [2,] 0.1245258 0.1241943 0.1266026
## [3,] 0.1251299 0.1246453 0.1267355
## [4,] 0.1257258 0.1250894 0.1268666
## [5,] 0.1263137 0.1255267 0.1269959
## [6,] 0.1268936 0.1259574 0.1271235
```

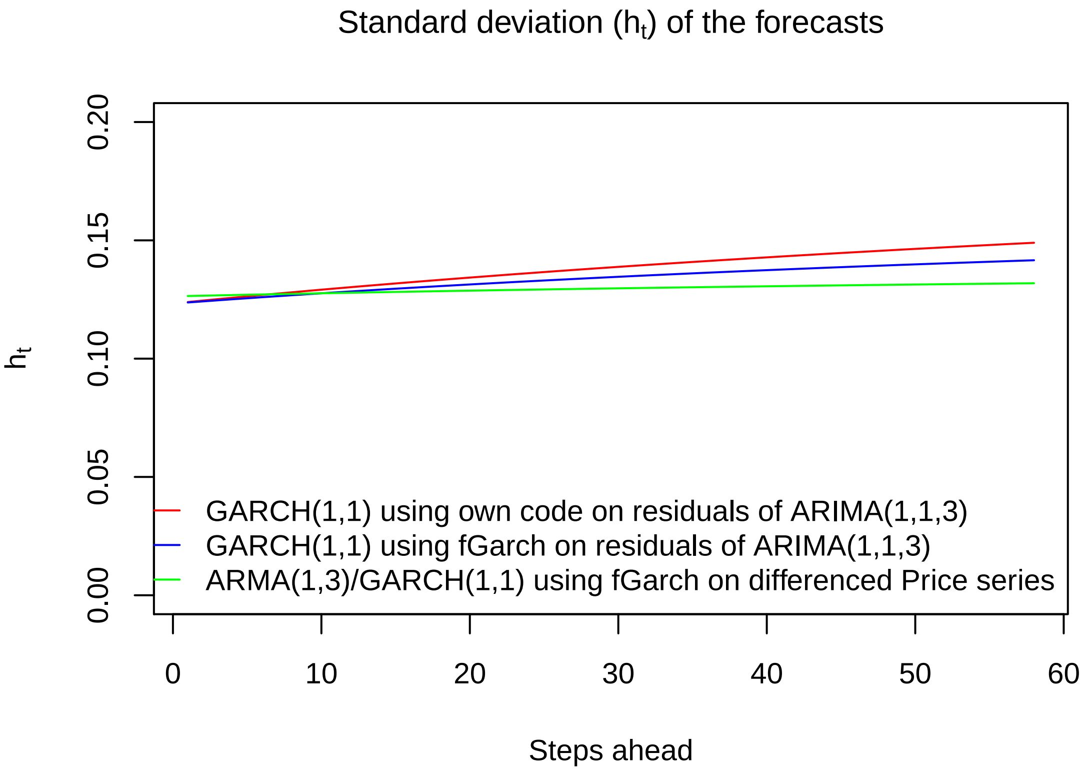


Figure 52: Standard deviation (h_t) of the U.S. inflation-adjusted average gas price forecasts for the 3 methods used