

Линейный регрессионный анализ

Прогнозирование коротких временных
рядов

Лекция 6

Предупреждение

Это не самый лучший метод для прогнозирования временных рядов

Лучше

- ▶ экспоненциальное сглаживание
- ▶ ARIMA

Но: регрессионный анализ хорош для прогнозирования коротких временных рядов, а таких рядов много.

Прогнозирование

Прогноз — это оценка будущего значения некоторой величины научными методами.

При прогнозировании предполагается, что факторы, определявшие поведение рассматриваемой величины в прошлом, не изменят своего действия и в будущем. Качественно составленный прогноз должен выявить тенденции прошлого и распространить их в будущее. Тогда, если прогноз разойдется с действительностью, это будет означать, что **в действие вступил новый, ранее неизвестный фактор**.

Количественные методы прогнозирования основываются на обработке массивов числовых данных и разделяются на:

1. причинно-следственные методы (в биржевой торговле: фундаментальный анализ);
2. методы анализа временных рядов (технический анализ).

Временные ряды

Временной ряд — последовательность значений некоторого показателя Y (например, объема продаж) во времени.

Развитие процессов, реально наблюдаемых в жизни, складывается из некоторой устойчивой тенденции (тренда, T) и случайной составляющей (ошибки, E), выражающейся в колебании значений показателя вокруг тренда. Зачастую временной ряд характеризуется также сезонными (S) или циклическими составляющими.

$$Y = T + S + E$$

Понятие сезона может трактоваться весьма широко: неделя, месяц, квартал, год или даже десятки лет.

Прогнозирование пассажирских авиаперевозок

Международные пассажирские авиаперевозки (`series_g.csv`).

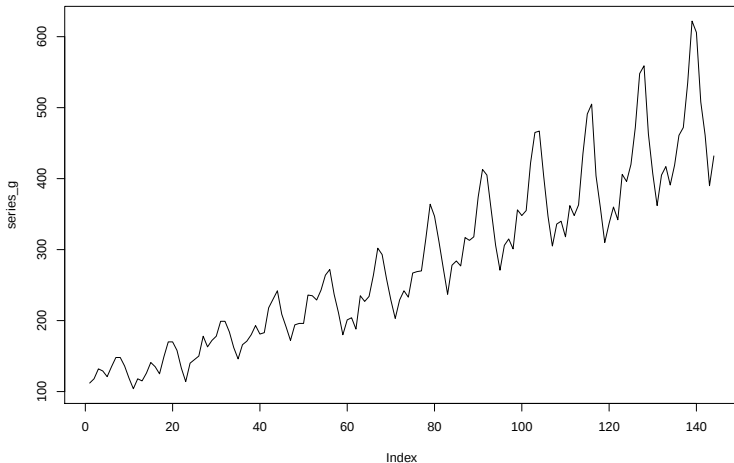
Переменные:

1. `date` — Даты, по месяцам: с января 1949 по декабрь 1960 года.
2. `series_g` — Объем пассажирских авиаперевозок, в тысячах человек.

Построим график временного ряда и зададим себе 4 вопроса:

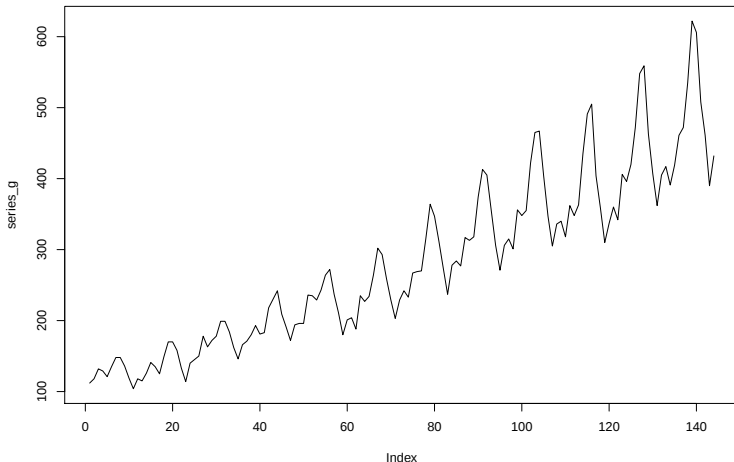
1. Есть ли у ряда тренд?
2. Есть ли сезонность?
3. Меняет ли ряд свой характер?
4. Есть ли в данных выбросы?

1. Есть ли у ряда тренд?



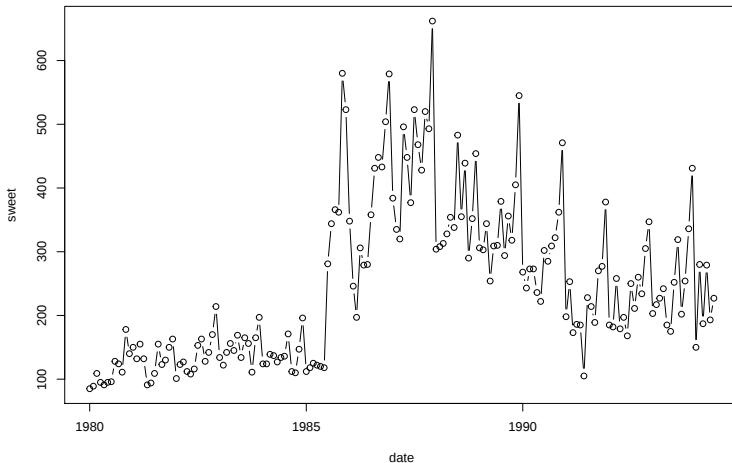
- Можем ли мы считать тренд линейным? Нелинейным?

2. Есть ли у ряда сезонность?



- Если сезонность есть, то к какому виду она относится: аддитивному или мультипликативному?

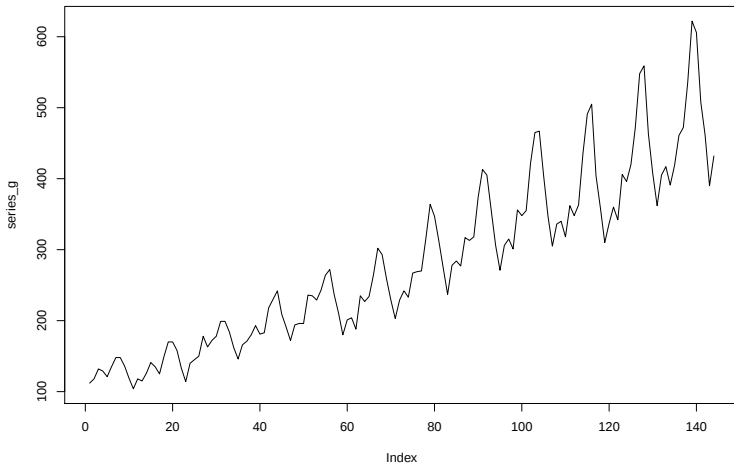
Меняет ли ряд свой характер?



Пример ряда, меняющего свой характер (потребление ликеров в Австралии).

График помогает отрезать данные, которые уже устарели.

4. Есть ли в данных выбросы?



Выбросы можно заменять на более разумные значения.

Последовательность работы над прогнозом

Построить график и задать себе 4 вопроса:

1. Есть ли у ряда тренд?
 - ▶ Можем ли мы аналитически описать этот тренд? (линейный, нелинейный)
2. Есть ли сезонность?
 - ▶ Если сезонность есть, то к какому виду она относится: аддитивному или мультипликативному?
3. Меняет ли ряд свой характер? (Помогает отрезать данные, которые уже устарели)
4. Есть ли в данных выбросы? (Их можно заменять на более разумные значения)

У нас

- ▶ Тренд есть, тренд линейный.
- ▶ Сезонность есть, мультипликативная.
- ▶ Ряд характер не меняет.
- ▶ Выбросов не наблюдается.

Но: у нас

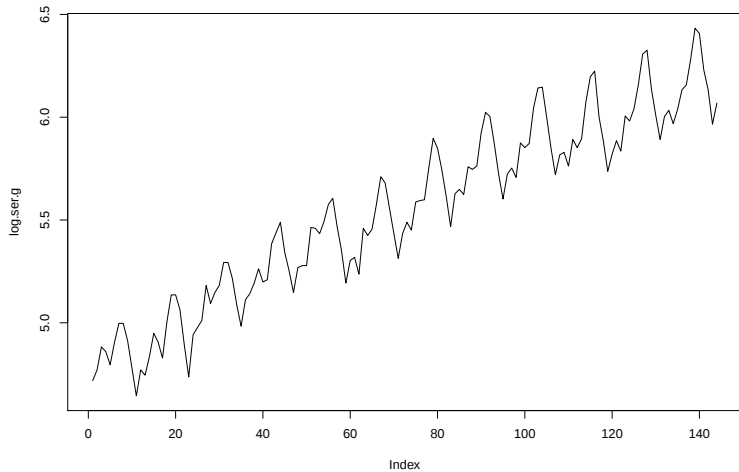
$$Y = T \cdot S \cdot E,$$

тогда как уравнение регрессии

$$Y = b_0 + \sum_{i=1}^N b_i X_i$$

Чтобы получить аддитивную сезонность, логарифмируем ряд.

Логарифм временного ряда



Индикаторные переменные (dummy variables)

Нужны чтобы моделировать сезонность (12 месяцев)

$$Y = b_0 + b_1T + \sum_{i=1}^{12} c_i D_i$$

D_i равна 1 для i -го месяца и 0 для остальных.

	log.ser.g	time.	month.01	month.02	month.03	month.04	... month.12
1	4.718499	1	1	0	0	0	0
2	4.770685	2	0	1	0	0	0
3	4.882802	3	0	0	1	0	0
4	4.859812	4	0	0	0	1	0
5	4.795791	5	0	0	0	0	0
6	4.905275	6	0	0	0	0	0
7	4.997212	7	0	0	0	0	0
8	4.997212	8	0	0	0	0	0
9	4.912655	9	0	0	0	0	0
10	4.779123	10	0	0	0	0	0
11	4.644391	11	0	0	0	0	0
12	4.770685	12	0	0	0	0	1
13	4.744932	13	1	0	0	0	0
...							

Пробуем

```
res.01 <- lm(log.ser.g ~ time. + month.01 + month.02 +  
              month.03 + month.04 + month.05 + month.06 +  
              month.07 + month.08 + month.09 + month.10 +  
              month.11 + month.12, ser.g.02)  
summary(res.01)
```

Coefficients: (1 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	4.7054593	0.0194850	241.491	< 2e-16	***
time.	0.0100688	0.0001193	84.399	< 2e-16	***
month.01	0.0213211	0.0242461	0.879	0.380816	
month.02	-0.0007338	0.0242400	-0.030	0.975897	
month.03	0.1294934	0.0242344	5.343	3.92e-07	***
month.04	0.0982245	0.0242294	4.054	8.59e-05	***
month.05	0.0958519	0.0242250	3.957	0.000124	***
month.06	0.2179981	0.0242212	9.000	2.25e-15	***
month.07	0.3219404	0.0242179	13.293	< 2e-16	***
month.08	0.3126456	0.0242153	12.911	< 2e-16	***
month.09	0.1680110	0.0242132	6.939	1.64e-10	***
month.10	0.0298527	0.0242118	1.233	0.219790	
month.11	-0.1138650	0.0242109	-4.703	6.41e-06	***
month.12	NA	NA	NA	NA	

Ловушка индикаторных переменных (dummy variables trap)

или уже знакомая коллинеарность.

$$Y = b_0 + b_1T + \sum_{i=1}^{12} c_i D_i$$

В матричном виде

$$\begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_{144} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & \dots & 0 \\ 1 & 2 & 0 & 1 & \dots & 0 \\ \dots & & & \dots & & \dots \\ 1 & 12 & 0 & 0 & \dots & 1 \\ 1 & 13 & 1 & 0 & \dots & 0 \\ \dots & & & \dots & & \dots \\ 1 & 144 & 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ c_1 \\ c_2 \\ \dots \\ c_{12} \end{bmatrix}$$

Складывая столбцы с $c_1, \dots, c_{12}$, получим столбец из одних 1, совпадающий со столбцом для b_0 . **Налицо линейная зависимость!**

Решение проблемы

Будем рассматривать не $N = 12$ индикаторных переменных (по числу сезонов), а $N - 1 = 11$. Тогда коэффициенты при индикаторных переменных приобретут вид $c_i - c_1$ и будут рассматриваться как значения относительно базового месяца (например, января).

$$Y = (b_0 - c_1) + b_1T + \sum_{i=2}^{12} (c_i - c_1)D_i$$

Берем за базу берем январь

```
res.01 <- lm(log.ser.g ~ time. + month.02 +  
              month.03 + month.04 + month.05 + month.06 +  
              month.07 + month.08 + month.09 + month.10 +  
              month.11 + month.12, ser.g.02)  
  
# Просмотр результатов  
summary(res.01)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	4.7267804	0.0188935	250.180	< 2e-16	***
time.	0.0100688	0.0001193	84.399	< 2e-16	***
month.02	-0.0220548	0.0242109	-0.911	0.36400	
month.03	0.1081723	0.0242118	4.468	1.69e-05	***
month.04	0.0769034	0.0242132	3.176	0.00186	**
month.05	0.0745308	0.0242153	3.078	0.00254	**
month.06	0.1966770	0.0242179	8.121	2.98e-13	***
month.07	0.3006193	0.0242212	12.411	< 2e-16	***
month.08	0.2913245	0.0242250	12.026	< 2e-16	***
month.09	0.1466899	0.0242294	6.054	1.39e-08	***
month.10	0.0085316	0.0242344	0.352	0.72537	
month.11	-0.1351861	0.0242400	-5.577	1.34e-07	***
month.12	-0.0213211	0.0242461	-0.879	0.38082	

В R всё проще... Создаём независимые переменные

```
# Время
time <- 1:144
# Сезонные индикаторы
month <- as.factor(rep(1:12,12))
# Объединяем результаты в таблицу
ser.g.02 <- data.frame(log.ser.g, time, month)
# Убеждаемся, что сезонные индикаторы заданы фактором.
# Иначе не избежать ловушки индикаторных переменных
class(ser.g.02$month)
```

	log.ser.g	time	month
1	4.718499	1	1
2	4.770685	2	2
3	4.882802	3	3
4	4.859812	4	4
5	4.795791	5	5
6	4.905275	6	6
7	4.997212	7	7
8	4.997212	8	8
9	4.912655	9	9
10	4.779123	10	10

Строим линейную регрессионную модель

```
res.01 <- lm(log.ser.g ~ ., ser.g.02)
summary(res.01)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	4.7267804	0.0188935	250.180	< 2e-16	***
time	0.0100688	0.0001193	84.399	< 2e-16	***
month2	-0.0220548	0.0242109	-0.911	0.36400	
month3	0.1081723	0.0242118	4.468	1.69e-05	***
month4	0.0769034	0.0242132	3.176	0.00186	**
month5	0.0745308	0.0242153	3.078	0.00254	**
month6	0.1966770	0.0242179	8.121	2.98e-13	***
month7	0.3006193	0.0242212	12.411	< 2e-16	***
month8	0.2913245	0.0242250	12.026	< 2e-16	***
month9	0.1466899	0.0242294	6.054	1.39e-08	***
month10	0.0085316	0.0242344	0.352	0.72537	
month11	-0.1351861	0.0242400	-5.577	1.34e-07	***
month12	-0.0213211	0.0242461	-0.879	0.38082	

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

Residual standard error: 0.0593 on 131 degrees of freedom

Multiple R-squared: 0.9835, Adjusted R-squared: 0.982

Подгонка для логарифма ряда

```
# Нарисуем линии: красный - подгонка, зеленый - ряд  
plot(res.01$fitted.values, type="l", col="red")  
lines(ser.g.02$log.ser.g, col="green")
```

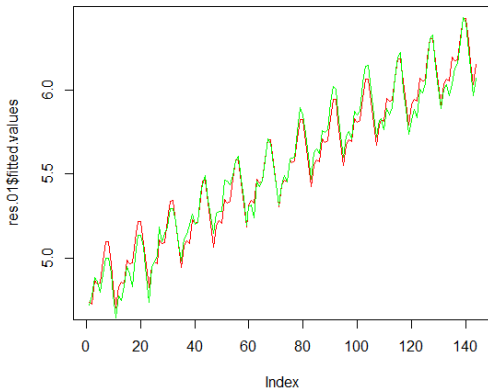
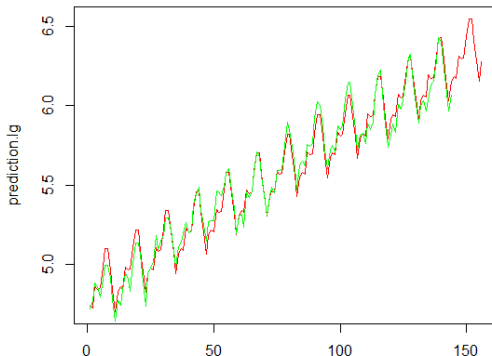


Figure 1: Вывод. Подгонка удовлетворительная. Можно надеяться на хороший прогноз

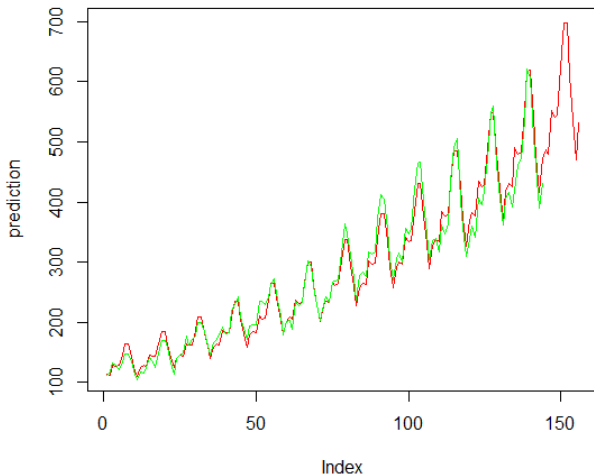
Прогнозирование логарифма ряда

```
# Создаем таблицу для новых значений  
ser.g.03 <- data.frame(time=145:156, month=factor(1:12))  
# Делаем прогноз при помощи модели res.01  
прогноз.lg = predict.lm(res.01, ser.g.03)  
# Объединяем подгонку и прогноз  
prediction.lg <- c(res.01$fitted.values, прогноз.lg)  
# Выводим на график  
plot(prediction.lg, type="l", col="red")  
lines(ser.g.02$log.ser.g, col="green")
```



Прогноз

```
# Потенцируем результат  
prediction <- exp(prediction.lg)  
# Выводим результат и прогноз  
plot(prediction, type="l", col="red")  
lines(ser.g.01$series_g, col="green")
```



Вернемся к таблице коэффициентов

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	4.7267804	0.0188935	250.180	< 2e-16	***
time	0.0100688	0.0001193	84.399	< 2e-16	***
month2	-0.0220548	0.0242109	-0.911	0.36400	
month3	0.1081723	0.0242118	4.468	1.69e-05	***
month4	0.0769034	0.0242132	3.176	0.00186	**
month5	0.0745308	0.0242153	3.078	0.00254	**
month6	0.1966770	0.0242179	8.121	2.98e-13	***
month7	0.3006193	0.0242212	12.411	< 2e-16	***
month8	0.2913245	0.0242250	12.026	< 2e-16	***
month9	0.1466899	0.0242294	6.054	1.39e-08	***
month10	0.0085316	0.0242344	0.352	0.72537	
month11	-0.1351861	0.0242400	-5.577	1.34e-07	***
month12	-0.0213211	0.0242461	-0.879	0.38082	

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

Достаточно ли данных?

Эмпирическое правило: чтобы данных было достаточно (например, для качественного прогноза) на каждую переменную должно приходиться 30 наблюдений.

Исходят из того, что 30 наблюдений достаточно для хорошего представления данных, распределенных по нормальному закону.

У нас 12 переменных (время плюс сезонные поправки), нам нужно $12 \cdot 30 = 360$ наблюдений, а есть всего 144.

Декабрьскую, январскую и февральскую сезонные поправки можно объединить в “зимнюю” поправку.

Объединение улучшит модель, так как на одну переменную будет приходиться больше наблюдений. Теперь их нужно только $10 \cdot 30 = 300$, а не 360.

Продажи вин в Австралии

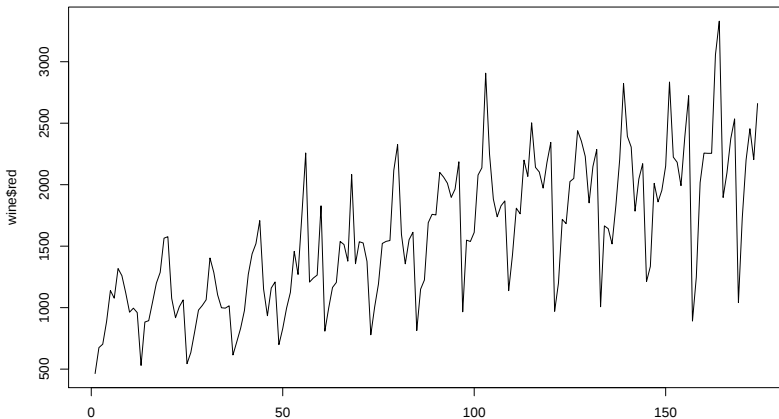
Ежемесячные данные с января 1980 по июнь 1994.

- ▶ fort - крепленые (тысячи литров)
- ▶ dry - сухие вина (тысячи литров)
- ▶ sweet - сладкие вина (тысячи литров)
- ▶ red - красные вина (тысячи литров)
- ▶ rose - розовые вина (тысячи литров)
- ▶ spark - игристые вина (тысячи литров)
- ▶ total - общие продажи вин производителями в бутылках объемом не более одного литра

Необходимо построить прогноз на 8 месяцев.

Читаем данные и строим график

```
# Шаг 0. Прочитаем данные. Внимание: разделитель полей Tab!  
wine <- read.table("wine_Austral.dat", header=T, sep="\t") #  
Шаг 1. Предварительный анализ: построим график ряда  
plot(wine$red, type="l")
```



Результаты анализа

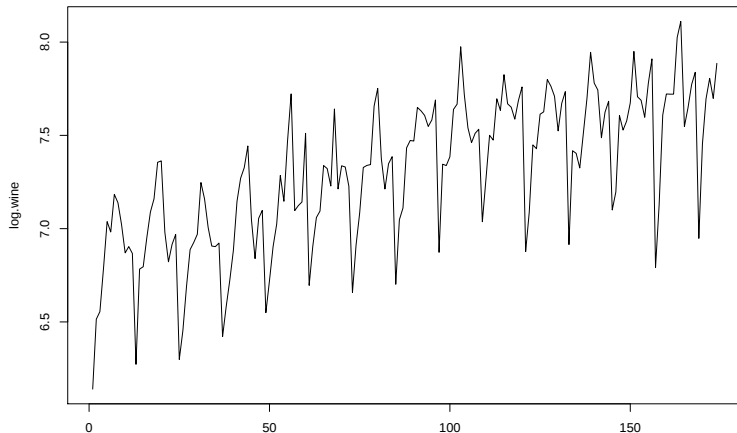
Из графика видно, что:

- ▶ Тренд есть, тренд линейный.
- ▶ Сезонность есть, мультипликативная.

Чтобы получить аддитивную сезонность, можно попробовать рассмотреть логарифм ряда.

Шаг 2. Преобразование временного ряда

```
log.wine <- log(wine$red)  
# Посмотрим результат на графике  
plot(log.wine, type="l")
```



Анализ логарифмированных данных

- ▶ Тренд есть, тренд примерно линейный.
- ▶ Сезонность есть, аддитивная.

Выводы:

- ▶ Преобразование привело к желаемому результату.
- ▶ Можно строить регрессионную модель с линейным трендом

Шаг 3. Создание дополнительных переменных

Создаем независимые переменные. Делаем это с запасом на те месяцы, для которых будет строиться прогноз

```
len <- nrow(wine)+8 # Нужно 175+8 строк
# Время
time <- 1:len
# Сезонные индикаторы
month <- as.factor(rep_len(1:12, len))
# Чтобы уравнивать длины всех векторов
# добавим к исходным данным пропущенные значения
log.wine[175:len] <- NA
# Для удобства работы склеиваем из векторов таблицу
wine.02 <- data.frame(log.wine, time, month)
```

Шаг 4. Регрессионный анализ

```
# Линейная регрессия. За базу автоматически берется январь
res.01 <- lm(log.wine ~ . , wine.02)
# Просмотр результатов
summary(res.01)
```

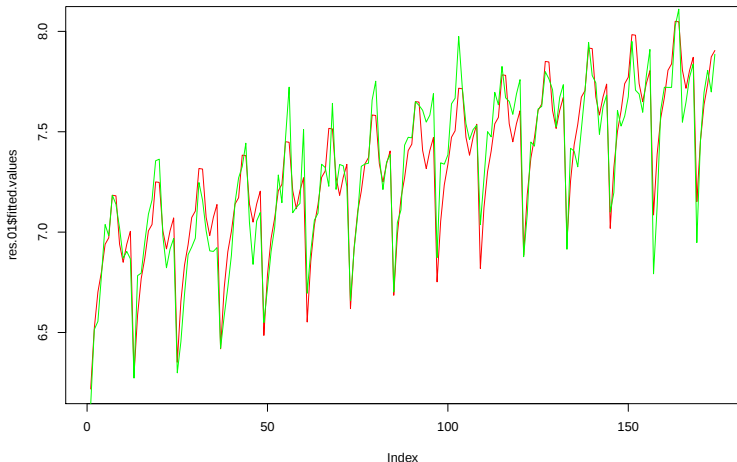
Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	6.2131856	0.0348337	178.37	< 2e-16	***
time	0.0055557	0.0001824	30.45	< 2e-16	***
month2	0.2968686	0.0441120	6.73	2.83e-10	***
month3	0.4719070	0.0441131	10.70	< 2e-16	***
month4	0.5686400	0.0441150	12.89	< 2e-16	***
month5	0.6984818	0.0441176	15.83	< 2e-16	***
month6	0.7248984	0.0441210	16.43	< 2e-16	***
month7	0.9315212	0.0448924	20.75	< 2e-16	***
month8	0.9233443	0.0448928	20.57	< 2e-16	***
month9	0.6769936	0.0448939	15.08	< 2e-16	***
month10	0.5809970	0.0448957	12.94	< 2e-16	***
month11	0.6657786	0.0448983	14.83	< 2e-16	***
month12	0.7244539	0.0449017	16.13	< 2e-16	***

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

Шаг 5. Подгонка для логарифма ряда

```
plot(res.01$fitted.values, type="l", col="red")  
lines(wine.02$log.wine, col="green")
```

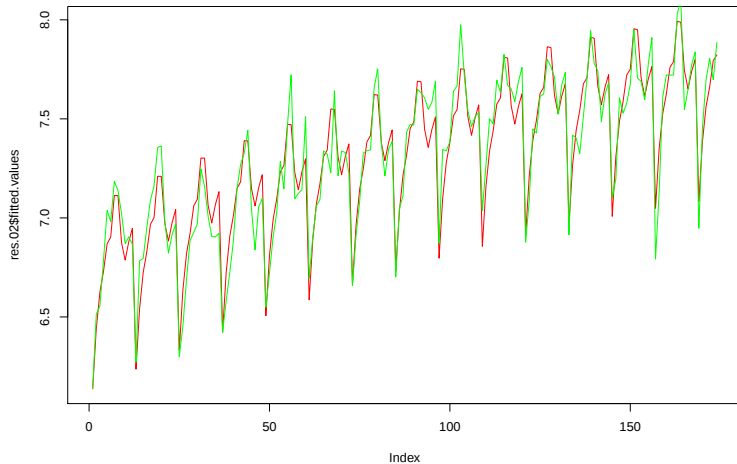


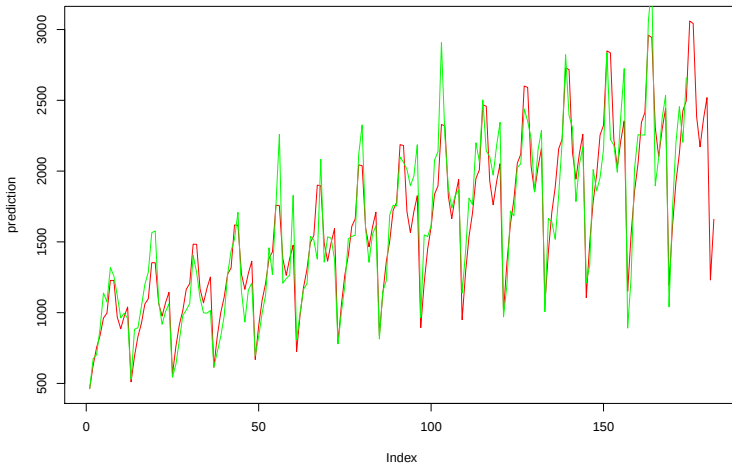
Попытка с параболическим трендом

```
# Время
time <- 1:len
time2 <- time*time
...
wine.03 <- data.frame(log.wine, time, time2, month)
```

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	6.128e+00	3.811e-02	160.795	< 2e-16	***
time	8.549e-03	6.960e-04	12.283	< 2e-16	***
time2	-1.711e-05	3.853e-06	-4.440	1.67e-05	***
month2	2.968e-01	4.175e-02	7.109	3.67e-11	***
month3	4.718e-01	4.175e-02	11.300	< 2e-16	***
month4	5.685e-01	4.176e-02	13.616	< 2e-16	***
month5	6.984e-01	4.176e-02	16.725	< 2e-16	***
month6	7.249e-01	4.176e-02	17.358	< 2e-16	***
month7	9.256e-01	4.251e-02	21.772	< 2e-16	***
month8	9.173e-01	4.251e-02	21.577	< 2e-16	***
month9	6.709e-01	4.251e-02	15.781	< 2e-16	***
month10	5.749e-01	4.252e-02	13.523	< 2e-16	***
month11	6.598e-01	4.252e-02	15.517	< 2e-16	***
month12	7.185e-01	4.252e-02	16.898	< 2e-16	***

График с параболическим трендом





Возможно следует отрезать первые 4 года наблюдений и это позволит точнее прогнозировать размах сезонных колебаний на оставшейся части ряда.