Lab_5

Могильников Дмитрий

2022-12-17

Задание 1

Сформируйте датасет самостоятельно на основе погодных данных (например, с сайта rp5).

С сайта rp5 были взяты данные города Нижневартовск за последние пять лет. Произведен парсинг этих данных в отдельный csv файл, в котором выделены два столбца:

- Дата
- Минимальная температура за сутки

Загрузим полученный датасет:

```
options(width = 100)
library(forecast)
```

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

```
library(lubridate)
```

```
## Загрузка требуемого пакета: timechange
```

```
##
## Присоединяю пакет: 'lubridate'
```

```
## Следующие объекты скрыты от 'package:base':
##
##     date, intersect, setdiff, union
```

```
df_min_temp <- read.csv('Nizhnevartovsk_min_temperature.csv', sep=',', header = TRUE)
head(df_min_temp, 20)
```

```
##          date min_temperature
## 1  2017-12-17            -9.8
## 2  2017-12-18           -12.7
## 3  2017-12-19            -9.1
## 4  2017-12-20           -13.3
## 5  2017-12-21           -10.5
## 6  2017-12-22            -9.3
## 7  2017-12-23            -8.0
## 8  2017-12-24            -9.4
## 9  2017-12-25           -15.0
## 10 2017-12-26           -19.9
## 11 2017-12-27           -24.4
## 12 2017-12-28           -26.4
## 13 2017-12-29           -19.6
## 14 2017-12-30           -23.9
## 15 2017-12-31           -27.0
## 16 2018-01-01           -13.7
## 17 2018-01-02           -20.2
## 18 2018-01-03           -36.1
## 19 2018-01-04           -25.4
## 20 2018-01-05            -8.4
```

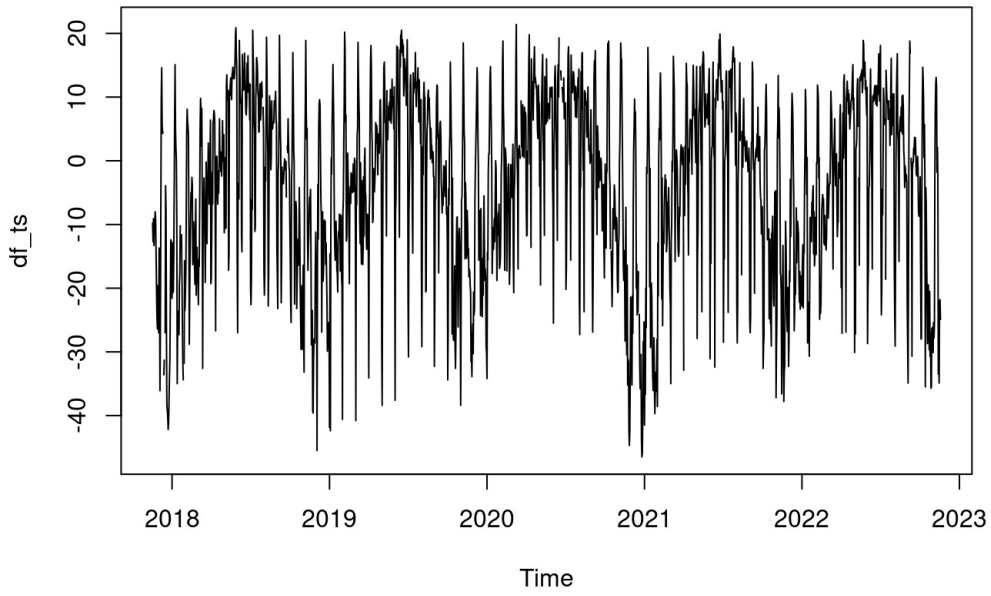
```
#Спарсим первую колонку в правильное значение даты
df_min_temp$date <- as.Date(df_min_temp$date, "%Y-%m-%d")
```

Задание 2

Создайте на основе датасета одномерный временной ряд. Выведите его на графике.

```
#Сформируем time-series, начиная с 2017-11-17
df_ts = ts(df_min_temp$min_temperature,
           frequency=365,
           start=decimal_date(ymd("2017-11-17")),
)

#Построим временной ряд:
plot(df_ts)
```




```
# Видим, что наблюдается сезонная периодичность, что естественно

#создадим переменную с датами
time <- time(df_ts)

#разделим на тестовую (15% ~270 элементов) и тестовую выборку
n_test <- 270
n_train <- length(df_ts) - n_test
df_train <- window(df_ts, start=time[1], end=time[n_train])
df_test <- window(df_ts, start=time[n_train+1], end=time[n_train+n_test])
```

Задание 3

Смоделируйте ряд тремя разными методами на Ваш выбор (naïve, snaive, ar, ma, arima, ses и т.д.).

```
# библиотека для проверки accuracy построенных моделей
library(knitr)
# Смоделируем наивную модель, h - сколько значений хотим предсказать(для сравнения с тестовой выборкой)
df_naive <- naive(df_train, h=n_test)

# Построим сезонную модель
df_snaive <- snaive(df_train, h=n_test)

# Рассчитаем accuracy для каждой модели
acc_naive <- accuracy(df_naive, df_test)
acc_snaive <- accuracy(df_snaive, df_test)

# Воспользуемся методом ARIMA: для начала рассмотрим автоматическую регрессию
df_ar <- arima(df_train, c(1,0,0))
df_ar_res <-predict(df_ar, n.ahead=n_test)
acc_ar <- accuracy(df_ar_res$pred, df_test)

# Рассмотрим скользящее среднее
df_ma <- arima(df_train, c(0,0,1))
df_ma_res <-predict(df_ma, n.ahead=n_test)
acc_ma <- accuracy(df_ma_res$pred, df_test)

# Рассмотрим полную ARIMA
df_arima <- arima(df_train, c(1,1,1))
df_arima_res <-predict(df_arima, n.ahead=n_test)
acc_arima <- accuracy(df_arima_res$pred, df_test)
```

Задание 4

Оцените точность прогнозирования построенных моделей (размер обучающей и тестовой выборок на Ваш выбор). Выберите наилучших метод.

```
# Сравним полученные результаты:
# Наивная модель
kable(acc_naive)
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	-0.0019506	9.544966	6.578284	NaN	Inf	0.8217388	-0.123298	NA
Test set	11.5165414	18.010740	16.355639	128.7663	424.4798	2.0430957	0.728016	1.955996

```
# Сезонная модель
kable(acc_snaive)
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	0.2049404	10.849156	7.999489	NaN	Inf	0.9992714	0.4754047	NA
Test set	-0.1612782	7.737141	5.843985	-12.07144	122.0261	0.7300125	0.3315575	0.6505751

```
# Авторегрессионная модель
kable(acc_ar)
```

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
Test set	3.182797	14.15237	12.01099	108.5145	186.6458	0.7268222	1.146815

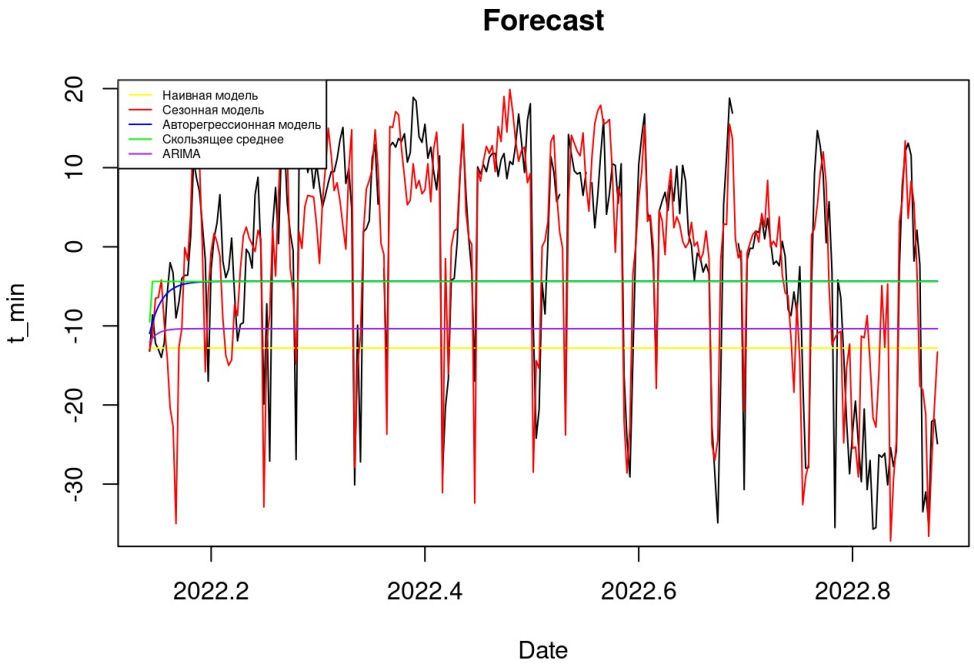
```
# Скользящее среднее
kable(acc_ma)
```

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
Test set	3.118175	14.18165	12.08093	109.7035	187.573	0.7286398	1.148454

```
# Полная ARIMA
kable(acc_arima)
```

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
Test set	9.084532	16.54989	14.96706	123.0399	353.8403	0.7278388	1.678547

```
plot(df_test, main="Forecast", xlab="Date", ylab="t_min")
lines(df_naive$mean, col="yellow")
lines(df_snaive$mean, col="red")
lines(df_ar_res$pred, col="blue")
lines(df_ma_res$pred, col="green")
lines(df_arima_res$pred, col="purple")
legend("topleft",
       legend=c("Наивная модель", "Сезонная модель", "Авторегрессионная модель", "Скользящее среднее", "ARIMA"),
       col=c("yellow", "red", "blue", "green", "purple"), lty = 1:1, cex=0.5)
```



Исходя из полученных результатов, мы можем сделать вывод, что для прогнозирования погоды лучше всего подходит сезонная модель, это связанно с тем, что только она учитывает необходимые сезонные изменения. У остальных моделей показатели MAE и RMSE хуже, а также на графике видно, что для такого небольшого промежутка времени они все сходятся к константе. Сезонная модель же достаточно хорошо соотносится с тестовой выборкой.

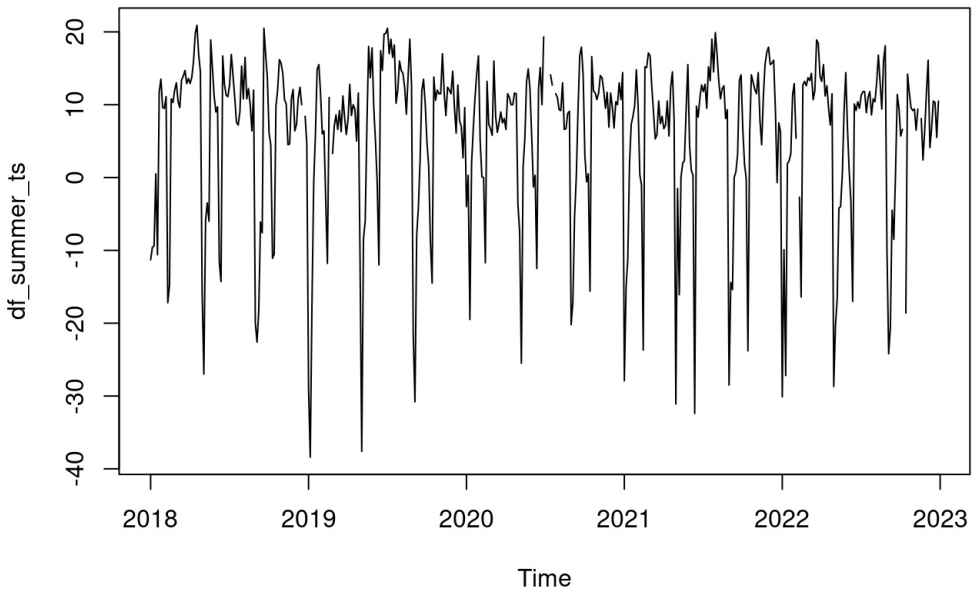
Задание 5

Сформируйте дополнительный датасет на основе первого, возьмите данные только за определенный сезон (весна, лето, осень, зима) минимально за последние 5 лет.

```
#Возьмем данные за лето, что поможет немного устранить сезонную компоненту.
df_summer <- (subset(df_min_temp, month(df_min_temp$date)>5 & month(df_min_temp$date)<9))

#Сформируем time-series, начиная с 2017-11-17
df_summer_ts = ts(df_summer$min_temperature,
           frequency=92,
           start = c(2018, 1)
)

#Построим временной ряд:
plot(df_summer_ts)
```




```
# Видим, что наблюдается сезонная периодичность, что естественно

#создадим переменную с датами
time_summer <- time(df_summer_ts)

#разделим на тестовую (15% ~60 элементов) и тестовую выборку
n_test_summer <- 60
n_train_summer <- length(df_summer_ts) - n_test_summer
df_train_summer <- window(df_summer_ts, start=time_summer[1], end=time_summer[n_train_summer])
df_test_summer <- window(df_summer_ts, start=time_summer[n_train_summer+1], end=time_summer[n_train_summer+n_test
_summer])
```

Задание 6

Те же пункты 2-4 ко второму датасету.

```
# Смоделируем наивную модель, h - сколько значений хотим предсказать(для сравнения с тестовой выборкой)
df_naive_summer <- naive(df_train_summer, h=n_test_summer)

# Построим сезонную модель
df_snaive_summer <- snaive(df_train_summer, h=n_test_summer)

# Рассчитаем accuracy для каждой модели
acc_naive_summer <- accuracy(df_naive_summer, df_test_summer)
acc_snaive_summer <- accuracy(df_snaive_summer, df_test_summer)

# Воспользуемся методом ARIMA: для начала рассмотрим автоматическую регрессию
df_ar_summer <- arima(df_train_summer, c(1,0,0))
df_ar_res_summer <-predict(df_ar_summer, n.ahead=n_test_summer)
acc_ar_summer <- accuracy(df_ar_res_summer$pred, df_test_summer)

# Рассмотрим скользящее среднее
df_ma_summer <- arima(df_train_summer, c(0,0,1))
df_ma_res_summer <-predict(df_ma_summer, n.ahead=n_test_summer)
acc_ma_summer <- accuracy(df_ma_res_summer$pred, df_test_summer)

# Рассмотрим полную ARIMA
df_arima_summer <- arima(df_train_summer, c(1,1,1))
df_arima_res_summer <-predict(df_arima_summer, n.ahead=n_test_summer)
acc_arima_summer <- accuracy(df_arima_res_summer$pred, df_test_summer)

# Сравним полученные результаты:
# Наивная модель
kable(acc_naive_summer)
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	-0.0229008	10.93955	6.692621	NaN	Inf	1.365187	-0.2172618	NA

Test set	25.7931034	27.54554	25.934483	380.9682	448.3189	5.290216	0.5136105	3.309321

```
# Сезонная модель
kable(acc_snaive_summer)
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	-0.1758278	7.129848	4.936755	NaN	Inf	1.007018	0.2609642	NA
Test set	-1.3327586	6.305375	4.791379	-33.3399	73.11252	0.977364	0.2759359	0.3927724

```
# Авторегрессионная модель
kable(acc_ar_summer)
```

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
Test set	0.0030318	9.127295	6.615603	18.5837	94.32836	0.5079397	0.7352167

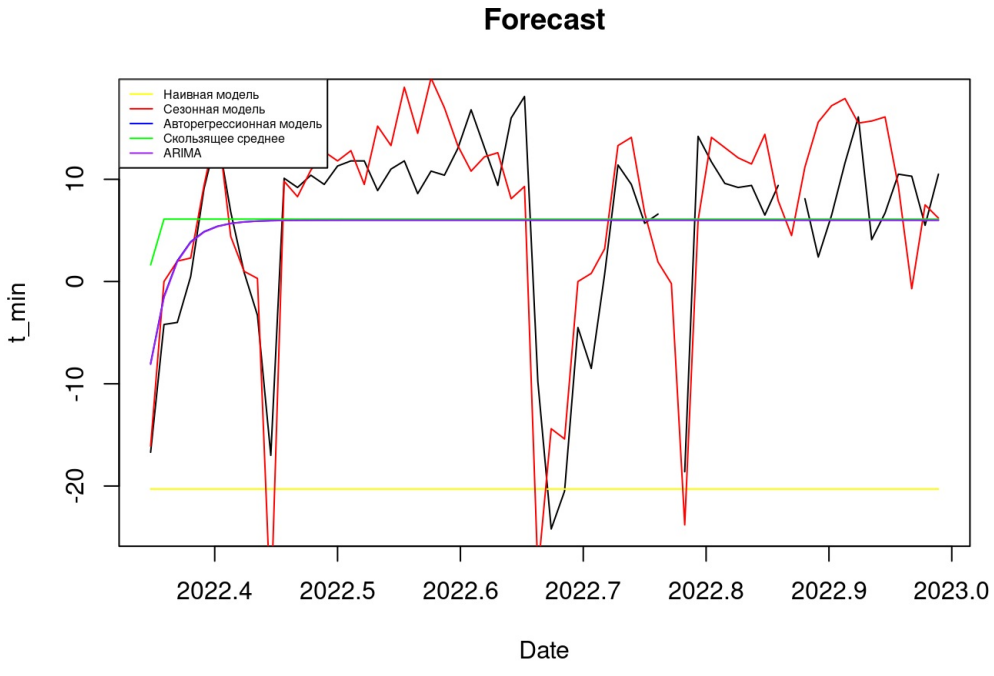
```
# Скользящее среднее
kable(acc_ma_summer)
```

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
Test set	-0.5364362	9.522379	6.944464	15.01676	107.9061	0.5220547	0.7030992

```
# Полная ARIMA
kable(acc_arima_summer)
```

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
Test set	0.0124203	9.124127	6.610899	18.75644	94.00066	0.5077359	0.7366301

```
plot(df_test_summer, main="Forecast", xlab="Date", ylab="t_min")
lines(df_naive_summer$mean, col="yellow")
lines(df_snaive_summer$mean, col="red")
lines(df_ar_res_summer$pred, col="blue")
lines(df_ma_res_summer$pred, col="green")
lines(df_arima_res_summer$pred, col="purple")
legend("topleft",
       legend=c("Наивная модель", "Сезонная модель", "Авторегрессионная модель", "Скользящее среднее", "ARIMA"),
       col=c("yellow", "red", "blue", "green", "purple"), lty = 1:1, cex=0.5)
```



Исходя из полученных результатов, мы можем сделать вывод, что наилучший результат прогнозирования погоды для одного сезона также дает сезонная модель. У остальных моделей показатели MAE и RMSE хуже, а также на графике видно, что для такого небольшого промежутка времени они все сходятся к константе. Сезонная модель же достаточно хорошо соотносится с тестовой выборкой.