Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного образовательного
учреждения высшего образования
**«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»**
(КФ МГТУ им. Н.Э. Баумана)

И.И. Ерохин

АНАЛИЗ И ПРОГНОЗИРОВАНИЕ ВРЕМЕННЫХ РЯДОВ. МЕТОД ARIMA.

Методические указания к выполнению лабораторной работы

по курсу «Технологии анализа данных»

Калуга – 2020

УДК 004.62
ББК 32.972.1
Б435

Методические указания составлены в соответствии с учебным планом КФ МГТУ им. Н.Э. Баумана по направлению подготовки 09.03.04 «Программная инженерия» кафедры «Программного обеспечения ЭВМ, информационных технологий».

Методические указания рассмотрены и одобрены:

- Кафедрой «Программного обеспечения ЭВМ, информационных технологий» (ИУ4-КФ) протокол № _ от «_ » _____ 2020 г.

Зав. кафедрой ИУ4-КФ _____к.т.н., доцент Ю.Е. Гагарин

- Методической комиссией факультета ИУ-КФ протокол №___ от «___» _____ 2020 г.

Председатель методической
комиссии факультета ИУ-КФ _____к.т.н., доцент М.Ю. Адкин

- Методической комиссией

КФ МГТУ им.Н.Э. Баумана протокол №___ от «___» _____ 2020 г.

Председатель методической
комиссии КФ МГТУ им.Н.Э.
Баумана _____д.э.н., профессор О.Л. Перерва

Рецензент:

к.т.н., доцент кафедры ИУ3-КФ _____ А.В. Финошин

Авторы

ассистент кафедры ИУ4-КФ _____ И.И. Ерохин

Аннотация

Методические указания к выполнению лабораторной работы по курсу «Технологии анализа данных» содержат общие сведения о библиотеках, применяемых в Python для анализа и прогнозирования временных рядов.

Предназначены для студентов 4-го курса бакалавриата КФ МГТУ им. Н.Э. Баумана, обучающихся по направлению подготовки 09.03.04 «Программная инженерия».

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Настоящие методические указания составлены в соответствии с программой проведения лабораторных работ по курсу «Технологии анализа данных» на кафедре «Программное обеспечение ЭВМ, информационные технологии» факультета «Информатика и управление» Калужского филиала МГТУ им. Н.Э. Баумана.

Методические указания, ориентированные на студентов 4-го курса направления подготовки 09.03.04 «Программная инженерия», содержат базовые сведения о библиотеках, применяемых в Python для анализа и прогнозирования временных рядов, а также принципах построения модели ARIMA для прогнозирования.

Методические указания составлены для ознакомления студентов с библиотеками, применяемые для анализа данных в языке Python. Для выполнения лабораторной работы студенту необходимы знания языка программирования Python и навыки работы с Anaconda.

ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ

Целью выполнения лабораторной работы является формирование практических навыков анализа и прогнозирования временных рядов, а также применения метода ARIMA.

Основными задачами выполнения лабораторной работы являются:

1. Ознакомиться с функциональными возможностями Python для анализа временных рядов.
2. Изучить метод ARIMA.

Результатами работы являются:

1. Визуализированные и проанализированные данные временных рядов.
2. Составленный прогноз.
3. Подготовленный отчет.

КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ

Временные ряды - важная разновидность структурированных данных. Они встречаются по многих областях, в том числе в финансах, экономике, экологии, нейронауках и физике. Любые результаты наблюдений или измерений в разные моменты времени, образуют временной ряд. Для многих временных рядов характерна фиксированном частота, т. е. интервалы между соседними точками одинаковы - измерения производятся, например, один раз в 15 секунд. 5 минут или в месяц. Но временные ряды могут быть и нерегулярными, когда интервалы времени между соседними точками различаются. Как разметить временной ряд и обращаться к нему, зависит от приложения. Существуют следующие виды временных рядов:

- временные метки, конкретные моменты времени;
- фиксированные периоды, например, январь 2007 или весь 2010 год;
- временные интервалы, обозначаемые метками начала и конца; Периоды можно считать частными случаями интервалов;
- время эксперимента или истекшее время. Каждая временная метка измеряет время, прошедшее с некоторого начального момента.

Временные ряды дают возможность прогнозировать будущие события. Основываясь на предыдущих значениях, временные ряды могут использоваться для прогнозирования тенденций в экономике, погоде, планировании производства и др. Специфические свойства данных временных рядов означают, что обычно требуются специальные статистические методы.

При анализе экономических временных рядов традиционно различают разные виды эволюции (динамики). Эти виды динамики могут, вообще говоря, комбинироваться. Тем самым задается разложение временного ряда на составляющие (компоненты), которые

с экономической точки зрения несут разную содержательную нагрузку. Наиболее важными из них являются:

Тенденция соответствует медленному изменению, проходящему в некотором определенном направлении, которое сохраняется в течение значительного промежутка времени. Тенденцию называют также трендом или долговременным движением.

Циклические колебания — это более быстрая, чем тенденция, квазипериодическая динамика, в которой есть фаза возрастания и фаза убывания. Наиболее часто цикл связан с флуктуациями экономической активности.

Сезонные колебания соответствуют изменениям, которые происходят регулярно в течение года, недели или суток. Они связаны с сезонами и ритмами человеческой активности.

Календарные эффекты — это отклонения, связанные с определенными предсказуемыми календарными событиями, такими как праздничные дни, количество рабочих дней за месяц, високосность года и т.п.

Случайные флуктуации — беспорядочные движения относительно большой частоты. Они порождаются влиянием разнородных событий на изучаемую величину (несистематический или случайный эффект).

Выбросы — это аномальные движения временного ряда, связанные с редко происходящими событиями, которые резко, но лишь очень кратковременно отклоняют ряд от общего закона, по которому он движется.

Структурные сдвиги — это аномальные движения временного ряда, связанные с редко происходящими событиями, имеющие скачкообразный характер и меняющие тенденцию.

В стандартной библиотеке Python имеются тины данных для представления даты и времени, а также средства, относящиеся к календарю (табл. 1):

Тип	**Описание**
date	Хранит дату (год. месяц, день) по

	григорианскому календарю
time	Хранит время суток (часы, минуты, секунды и микросекунды)
datetime	Хранит дату и время
timedelta	Представляет разность между двумя значениями типа datetime (дни, секунды и микросекунды)

Табл. 1. Типы данных в модуле datetime

Объекты типа datetime можно представить в виде отформатированной строки с помощью метода str или strftime, которому передается спецификация формата. Для обратного преобразования строк в даты используется метод datetime.strptime.

Метод datetime.strptime прекрасно работает, когда формат даты известен. Однако каждый раз задавать формат даты, особенно общеупотребительный, надоедает. В таком случае можно воспользоваться методом parser.parse из стороннего пакета dateutil, который умеет разбирать практически любое представление даты, понятное человеку.

Анализ временных рядов включает в себя методы анализа данных временных рядов с целью извлечения значимой статистики и других характеристик данных. Прогнозирование временных рядов - это использование модели для прогнозирования будущих значений на основе ранее наблюдаемых значений.

Один из методов, доступных в Python для моделирования и прогнозирования будущих точек временного ряда, известен как SARIMAX (Seasonal AutoRegressive Integrated Moving Averages with eXogenous regressors). В данном методическом указании будет рассмотрен компонент ARIMA, который используется для подгонки данных временных рядов для лучшего понимания и прогнозирования будущих значений во временных рядах.

ARIMA (AUTOREGRESSIVE INTEGRATED MOVING AVERAGE)

ARIMA (англ. autoregressive integrated moving average, иногда модель Бокса — Дженкинса, методология Бокса — Дженкинса) — интегрированная модель авторегрессии — скользящего среднего — модель и методология анализа временных рядов. Является расширением для нестационарных временных рядов.

ARIMA является одним из наиболее распространенных методов, используемых в прогнозировании временных рядов. ARIMA - это модель, которая может быть адаптирована к данным временного ряда, для лучшего понимания или предсказывания будущих значений в ряду.

Существует три различных целых числа (p, d, q), которые используются для параметризации моделей ARIMA. В связи с этим модели ARIMA обозначаются так: ARIMA (p, d, q). Вместе эти три параметра учитывают сезонность, тренд и шум в наборах данных:

p - авторегрессивная часть модели. Данный параметр позволяет включить влияние прошлых значений в модель. Интуитивно, это похоже на высказывание о том, что завтра может быть тепло, если было тепло в последние 3 дня.

d является неотъемлемой частью модели. Этот параметр включает в себя термины модели, которые содержат количество разностей (то есть количество прошлых временных точек, которые необходимо вычесть из текущего значения) для применения к временному ряду. Это похоже на утверждение о том, что завтра, вероятно, будет та же температура, если разница в температуре за последние три дня была очень мала.

q - скользящая средняя часть модели. Данный параметр позволяет представить ошибку модели как линейную комбинацию значений ошибок, наблюдаемых в предыдущие моменты времени.

При работе с сезонными эффектами используется сезонная ARIMA, которая обозначается как ARIMA (p, d, q) (P, D, Q) s. Здесь (p, d, q) - несезонные параметры, описанные выше, в то время как (P,

D, Q) следуют тому же определению, но применяются к сезонной компоненте временного ряда. Термин s - это периодичность временного ряда (4 - для квартальных периодов, 12 - для годовых периодов и т. д.).

Сезонный ARIMA может показаться сложным методом из-за множества параметров настройки. Далее будет показано, как автоматизировать процесс определения оптимального набора параметров для сезонной модели временных рядов ARIMA.

Выбор параметров для модели ARIMA

При поиске соответствия данных временного ряда сезонной модели ARIMA первой целью является поиск значений ARIMA (p, d, q) (P, D, Q), которые оптимизируют интересующую метрику. Для достижения этой цели существует множество руководств и рекомендаций, однако правильная параметризация моделей ARIMA может быть кропотливым ручным процессом, требующим специальных знаний и времени. Другие статистические языки программирования, такие как R, предоставляют автоматизированные способы решения этой проблемы. Эти способы необходимо перенести на Python. В данном разделе будут представлены теоретические сведения, а в следующем разделе - код на Python для программного выбора оптимальных значений параметров для модели временных рядов ARIMA (p, d, q) (P, D, Q).

Чтобы итеративно исследовать различные комбинации параметров, которые необходимо предварительно сгенерировать (каждый из параметров должен быть в диапазоне от 0 до 2), будет использован «поиск по сетке». Для каждой комбинации параметров подбирается новая модель ARIMA при помощи функции SARIMAX () из модуля statsmodels и оценивается ее общее качество. После того, как была изучена вся совокупность параметров, оптимальный набор параметров будет таким, который даст наилучшую производительность по интересующим критериям.

После генерации можно использовать тройки параметров, определенных выше, чтобы автоматизировать процесс обучения и

оценки моделей ARIMA для различных комбинаций. В статистике и машинном обучении этот процесс известен как поиск по сетке (или оптимизация гиперпараметров) для выбора модели.

При оценке и сравнении статистических моделей, снабженных различными параметрами, каждая из них может быть ранжирована на основании того, насколько хорошо она соответствует данным или ее способности точно прогнозировать будущие значения данных. В данном методическом указании будет использован коэффициент AIC (Akaike Information Criterion), который удобно подсчитывать моделями ARIMA, установленным с помощью statsmodels. AIC измеряет, насколько хорошо модель соответствует данным, принимая во внимание общую сложность модели. Модель, которая очень хорошо соответствует данным при использовании большого количества функций, получит более высокий балл AIC, чем модель, которая использует меньше функций для достижения того же качества соответствия. Поэтому нужно осуществить поиске модели, которая дает самое низкое значение AIC.

Функция SARIMAX из statsmodels для создания соответствующей сезонной модели ARIMA имеет несколько параметров:

- аргумент order указывает на параметры (p, d, q),
- аргумент season_order указывает на сезонный компонент (P, D, Q, S) сезонной модели ARIMA.

Поскольку некоторые комбинации параметров могут привести к ошибочным ситуациям, необходимо явно отключить предупреждающие сообщения, чтобы избежать перегрузки. Неправильные параметры могут также привести к ошибкам и исключениям, поэтому необходимо перехватить эти исключения и проигнорировать комбинации параметров, которые их вызывают.

ОБРАЗЕЦ ВЫПОЛНЕНИЯ ЗАДАНИЯ

```
import warnings
import itertools
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import statsmodels.api as sm
import matplotlib
from pylab import rcParams

warnings.filterwarnings("ignore")
plt.style.use('fivethirtyeight')

matplotlib.rcParams['axes.labelsize'] = 14
matplotlib.rcParams['xtick.labelsize'] = 12
matplotlib.rcParams['ytick.labelsize'] = 12
matplotlib.rcParams['text.color'] = 'k'


df = pd.read_excel("data/lab3/Superstore.xls")
furniture = df.loc[df['Category'] == 'Furniture']

# удаление ненужных столбцов (остаются только столбцы Date и
Sales)
cols = ['Row ID', 'Order ID', 'Ship Date', 'Ship Mode',
        'Customer ID', 'Customer Name', 'Segment', 'Country',
        'City', 'State', 'Postal Code', 'Region', 'Product ID',
        'Category', 'Sub-Category', 'Product Name', 'Quantity',
        'Discount', 'Profit']
furniture.drop(cols, axis=1, inplace=True)

# сбрасываем индексы (начинаем нумерацию с 0) и устанавливаем
индекс на поле Date
furniture = furniture.sort_values('Order Date')
furniture = furniture.groupby('Order Date')
['Sales'].sum().reset_index()
furniture = furniture.set_index('Order Date')

# получение средних значений ежедневных продаж за месяц
# в качестве метки времени использовано начало каждого месяца
y = furniture['Sales'].resample('MS').mean()

y.plot(figsize=(15, 6))
plt.show()
```

```
   # можно визуализировать данные, используя метод,
   # называемый декомпозицией временных рядов, который позволяет
   # разбивать временной ряд на три отдельных компонента: тренд,
сезонность и шум
   rcParams['figure.figsize'] = 18, 10
   decomposition = sm.tsa.seasonal_decompose(y, model='additive')
   fig = decomposition.plot()
   plt.show()

   # График выше ясно показывает,
   # что продажи мебели нестабильны,
   # наряду с очевидной сезонностью.

   # задаем наборы параметров для ARIMA
   p = d = q = range(0, 2)
   pdq = list(itertools.product(p, d, q))
   seasonal_pdq = [(x[0], x[1], x[2], 12) for x in pdq]

   # выбираем наиболее подходящую модель (с наименьшим AIC)
   minAIC = float("inf")
   bestModel = None
   for param in pdq:
       for param_seasonal in seasonal_pdq:
           try:
               mod = sm.tsa.statespace.SARIMAX(y,
                                               order=param,
                                               seasonal_order=param_s
easonal,
                                               enforce_stationarity=F
alse,
                                               enforce_invertibility=
False)
               results = mod.fit()
               if(results.aic < minAIC ):
                   bestModel = mod
                   minAIC = results.aic
           except:
               continue

   results = bestModel.fit()

   # сравниваем прогноз с реальными данными
   pred  =  results.get_prediction(start=pd.to_datetime('2017-01-01'),
dynamic=False)
   pred_ci = pred.conf_int()
   ax = y['2014':].plot(label='Реальные данные')
```

```
   pred.predicted_mean.plot(ax=ax,     label='Прогноз',     alpha=.7,
figsize=(14, 7))
   ax.fill_between(pred_ci.index,
                   pred_ci.iloc[:, 0],
                   pred_ci.iloc[:, 1], color='k', alpha=.2)
   ax.set_xlabel('Дата')
   ax.set_ylabel('Продажи')
   plt.legend()
   plt.show()

   # вычисляем дисперсию и СКО
   y_forecasted = pred.predicted_mean
   y_truth = y['2017-01-01':]
   mse = ((y_forecasted - y_truth) ** 2).mean()
   print('Дисперсия = {}'.format(round(mse, 2)))
   print('СКО = {}'.format(round(np.sqrt(mse), 2)))

   # делаем прогноз на след. годы
   pred_uc = results.get_forecast(steps=100)
   pred_ci = pred_uc.conf_int()
   ax = y.plot(label='Реальные данные', figsize=(14, 7))
   pred_uc.predicted_mean.plot(ax=ax, label='Прогноз')
   ax.fill_between(pred_ci.index,
                   pred_ci.iloc[:, 0],
                   pred_ci.iloc[:, 1], color='k', alpha=.25)
   ax.set_xlabel('Дата')
   ax.set_ylabel('Продажи')
   plt.legend()
   plt.show()
```

В результате выполнения программы будут построены следующие графики (рис. 1, 2, 3, 4):

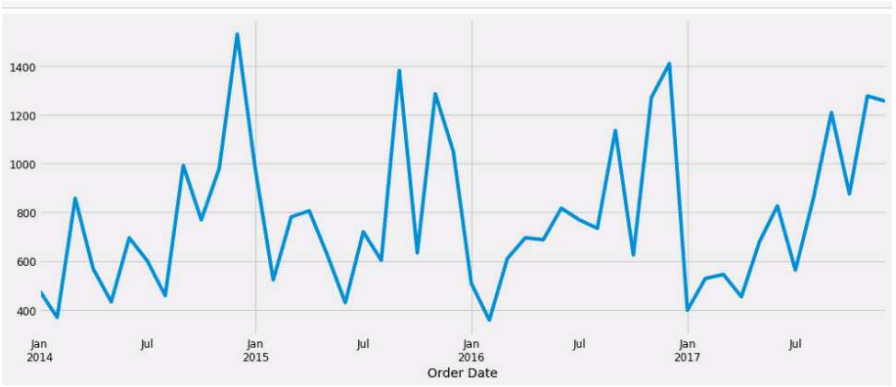


Рис. 1 График продаж

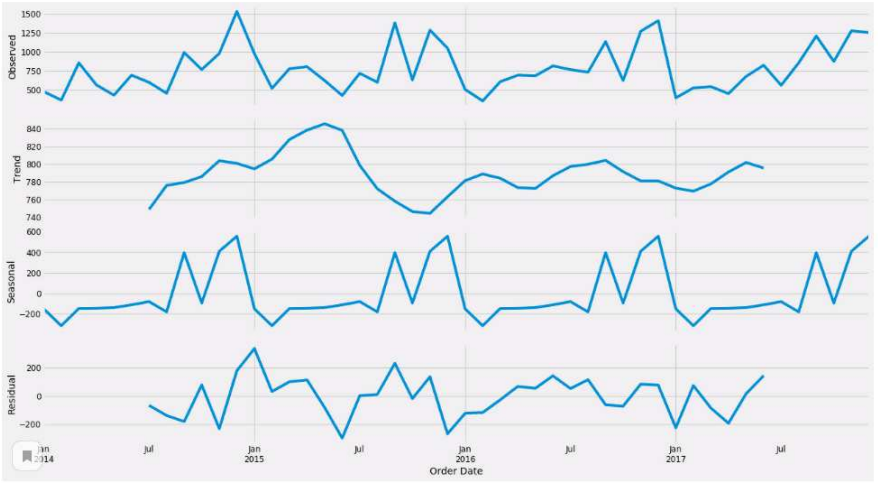
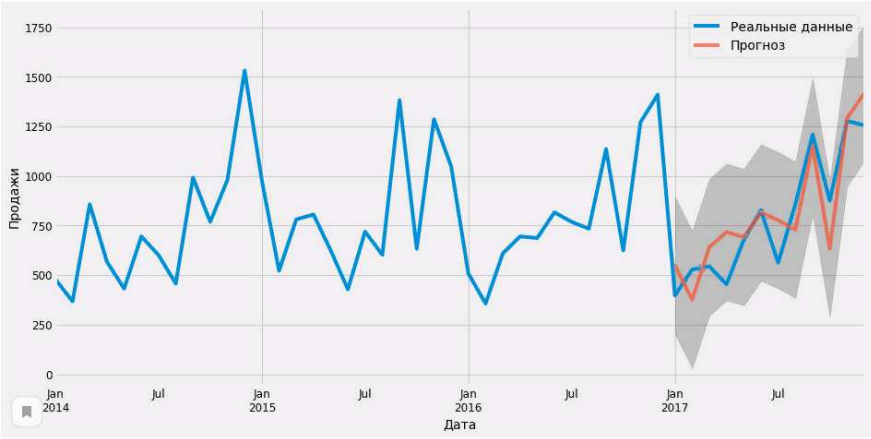


Рис. 2 Тренды, сезонность и шумы




```
Дисперсия = 22993.57
СКО = 151.64
```

Рис. 3 Сравнение прогноза с реальными данными

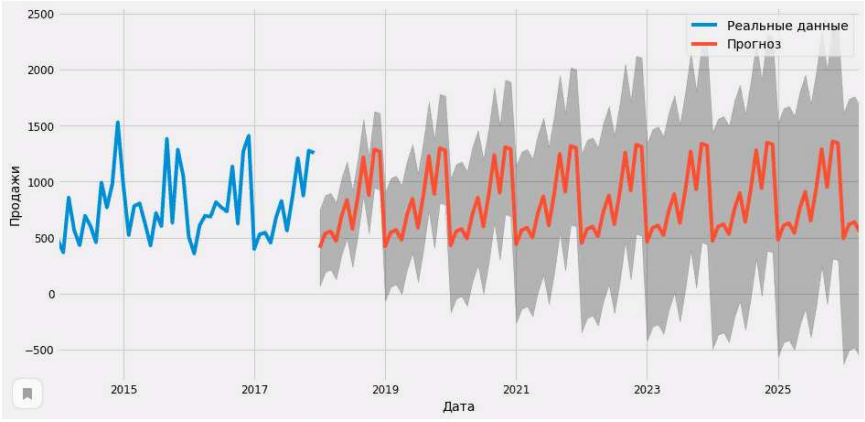


Рис.4 Прогнозирование на след. годы

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Для всех вариантов необходимо провести анализ временных рядов. Сделать выводы на основе данного анализа. Составить модель для прогнозирования по методу ARIMA и подобрать для нее параметры. Сравнить прогнозы полученной модели с реальными данными за период. При помощи построенной модели осуществить прогноз на будущие периоды. Данные находятся в файле Superstore.xls.

ТРЕБОВАНИЯ К РЕАЛИЗАЦИИ

В качестве результата работы необходимо построить графики и отобразить их в Jupyter Notebook. Также необходимо проанализировать полученные результаты и сделать выводы. Для чтения xls (exel) файла необходимо установить пакет xlrd.

ВАРИАНТЫ ЗАДАНИЙ

Вариант 1

Считать данные в структуру Dataframe. Выбрать из набора данные, относящиеся к категории Technology. Визуализировать продажи по месяцам (тренды, сезонность, шумы) для подкатегорий (2-3 подкатегории) и отобразить на одном графике. Сделать выводы по построенным графикам. Составить модель для прогноза по методу ARIMA и осуществить прогноз продаж на 5 лет по категории Technology в целом. Оценить качество модели (дисперсия и СКО от реальных данных за какой-либо период). Изобразить прогноз на графике.

Вариант 2

Считать данные в структуру Dataframe. Выбрать из набора данные, относящиеся к категории Office Supplies. Визуализировать продажи по месяцам (тренды, сезонность, шумы) для подкатегорий (2-3 подкатегории) и отобразить на одном графике. Сделать выводы по построенным графикам. Составить модель для прогноза по методу ARIMA и осуществить прогноз продаж на 5 лет по категории Office

Supplies в целом. Оценить качество модели (дисперсия и СКО от реальных данных за какой-либо период). Изобразить прогноз на графике.

Вариант 3

Считать данные в структуру Dataframe. Выбрать из набора данные, относящиеся к категориям Office Supplies и Technology. Визуализировать продажи по месяцам (тренды, сезонность, шумы) для обеих категорий и отобразить на одном графике. Сделать выводы по построенным графикам. Построить график СКО между продажами обеих категорий. Составить модель для прогноза по методу ARIMA и осуществить прогноз СКО между продажами обеих категорий на 5 лет. Оценить качество модели (дисперсия и СКО от реальных данных за какой-либо период). Изобразить прогноз на графике.

Вариант 4

Считать данные в структуру Dataframe. Выбрать из набора данные, относящиеся к подкатегориям Bookcases и Tables. Визуализировать продажи по месяцам (тренды, сезонность, шумы) для обеих категорий и отобразить на одном графике. Сделать выводы по построенным графикам. Построить график СКО между продажами обеих подкатегорий. Составить модель для прогноза по методу ARIMA и осуществить прогноз СКО между продажами обеих подкатегорий на 5 лет. Оценить качество модели (дисперсия и СКО от реальных данных за какой-либо период). Изобразить прогноз на графике.

Вариант 5

Считать данные в структуру Dataframe. Выбрать из набора данные, относящиеся ко всем подкатегориям. Определить какая из подкатегорий будет лучше всего продаваться через 5 лет на основе графических данных модели прогноза по методу ARIMA. Оценить качество модели (дисперсия и СКО от реальных данных за какой-либо период) на примере любых двух подкатегорий.

Вариант 6

Считать данные в структуру Dataframe. Выбрать из набора данные, относящиеся ко всем подкатегориям. Определить какая из подкатегорий будет хуже всего продаваться через 5 лет на основе графических данных модели прогноза по методу ARIMA. Оценить качество модели (дисперсия и СКО от реальных данных за какой-либо период) на примере любых двух подкатегорий.

Вариант 7

Считать данные в структуру Dataframe. Выбрать из набора данные, относящиеся ко всем подкатегориям. Определить какая из подкатегорий будет стабильнее всего продаваться на протяжении последующих 5 лет на основе графических данных модели прогноза по методу ARIMA. Оценить качество модели (дисперсия и СКО от реальных данных за какой-либо период) на примере любых двух подкатегорий.

Вариант 8

Считать данные в структуру Dataframe. Выбрать из набора данные, относящиеся к категории Office Supplies. Визуализировать продажи по месяцам (тренды, сезонность, шумы). Сделать выводы по построенным графикам. Составить модель для прогноза по методу ARIMA и осуществить прогноз СКО между продажами данной категории через 5 лет и продажами в текущем году. Оценить качество модели (дисперсия и СКО от реальных данных за какой-либо период). Изобразить прогноз на графике.

Вариант 9

Считать данные в структуру Dataframe. Выбрать из набора данные, относящиеся к категории Technology. Визуализировать продажи по месяцам (тренды, сезонность, шумы). Сделать выводы по построенным графикам. Составить модель для прогноза по методу ARIMA и осуществить прогноз СКО между продажами данной категории через 5 лет и продажами за первый год. Оценить качество

модели (дисперсия и СКО от реальных данных за какой-либо период). Изобразить прогноз на графике.

Вариант 10

Считать данные в структуру Dataframe. Выбрать из набора данные, относящиеся ко всем подкатегориям. На основе графических данных определить для каждой подкатегории год, в котором на нее был наибольший тренд. С помощью модели прогноза по методу ARIMA сделать прогноз продаж на 5 лет для любой подкатегории. Оценить качество модели (дисперсия и СКО от реальных данных за какой-либо период). Изобразить прогноз на графике.

Вариант 11

Считать данные в структуру Dataframe. Выбрать из набора данные, относящиеся к подкатегориям Storage и Art. Визуализировать продажи по месяцам (тренды, сезонность, шумы) для обеих категорий и отобразить на одном графике. Сделать выводы по построенным графикам. Построить график СКО между продажами обеих подкатегорий. Составить модель для прогноза по методу ARIMA и осуществить прогноз СКО между продажами обеих подкатегорий на 5 лет. Оценить качество модели (дисперсия и СКО от реальных данных за какой-либо период). Изобразить прогноз на графике.

Вариант 12

Считать данные в структуру Dataframe. Выбрать из набора данные, относящиеся ко всем подкатегориям. На основе графических данных определить категорию, которая продавалась стабильнее всего. С помощью модели прогноза по методу ARIMA сделать прогноз СКО между продажами на 5 лет для любых двух подкатегорий. Оценить качество модели (дисперсия и СКО от реальных данных за какой-либо период). Изобразить прогноз на графике.

Вариант 13

Считать данные в структуру Dataframe. Выбрать из набора данные, относящиеся к категории Furniture. Визуализировать продажи по месяцам (тренды, сезонность, шумы) для подкатегорий (2-3 подкатегории) и отобразить на одном графике. Сделать выводы по построенным графикам. Составить модель для прогноза по методу ARIMA и осуществить прогноз продаж на 5 лет для одной из подкатегорий. Оценить качество модели (дисперсия и СКО от реальных данных за какой-либо период). Изобразить прогноз на графике.

Вариант 14

Считать данные в структуру Dataframe. Выбрать из набора данные, относящиеся к категориям Office Supplies и Furniture. Визуализировать продажи по месяцам (тренды, сезонность, шумы) для обеих категорий и отобразить на одном графике. Сделать выводы по построенным графикам. Построить график СКО между продажами обеих категорий. Составить модель для прогноза по методу ARIMA и осуществить прогноз СКО между продажами обеих категорий на 5 лет. Оценить качество модели (дисперсия и СКО от реальных данных за какой-либо период). Изобразить прогноз на графике.

Вариант 15

Считать данные в структуру Dataframe. Выбрать из набора данные, относящиеся к подкатегориям Chairs и Tables. Визуализировать продажи по месяцам (тренды, сезонность, шумы) для обеих категорий и отобразить на одном графике. Сделать выводы по построенным графикам. Построить график СКО между продажами обеих подкатегорий. Составить модель для прогноза по методу ARIMA и осуществить прогноз продаж для обеих подкатегорий на 5 лет. Оценить качество модели (дисперсия и СКО (взять среднее арифметическое значений для каждой подкатегории) от реальных данных за какой-либо период). Изобразить прогнозы на одном графике.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Дайте определение понятию временной ряд.
2. Назовите два основных типа временных рядов.
3. Перечислите виды временных рядов.
4. Перечислите и раскройте основные экономические компоненты временных рядов.
5. Назовите типы данных, которые используются в Python для представления даты и времени.
6. Опишите, как осуществляется преобразование из строки в дату в Python.
7. Приведите сущность и назначение прогнозирования временных рядов.
8. Опишите суть метода ARIMA для анализа временных рядов.
9. Перечислите и раскройте параметры метода ARIMA.
10. Опишите процесс построения модели ARIMA.
11. Раскройте сущность коэффициента AIC.
12. Приведите примеры параметров, на основании которых можно судить о качестве модели ARIMA.

ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ

На выполнение лабораторной работы отводится 1 занятие (2 академических часа: 1 час на выполнение и сдачу лабораторной работы и 1 час на подготовку отчета).

Отчет на защиту предоставляется в печатном виде.

Структура отчета (на отдельном листе(-ах)): титульный лист, формулировка задания, описание процесса выполнения лабораторной работы, результаты выполнения работы, выводы.

ОСНОВНАЯ ЛИТЕРАТУРА

1. Маккинли, Уэс Python и анализ данных / Пер. с англ. Слинкин А.А. - М.: ДМК Пресс, 2015. - 482 с.:ил.
2. Грас, Дж. Data Science. Наука о данных с нуля / Пер. с англ. - СПБ.: БХВ -Петербург, 2017. - 336с.: ил.

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

3. Henley, A.J. Learn Data Analysis with Python: Lessons in Coding / A.J. Henley, Dave Wolf ISBN 978-1-4842-3486-0

Электронные ресурсы:

4. Научная электронная библиотека http://eLIBRARY.RU
5. Электронно-библиотечная система http://e.lanbook.com