

1. Введение в язык R

ПРЕИМУЩЕСТВА R

- + R – **свободный** пакет.
- + Активный процесс разработки ядра, **частые релизы**.
- + Сплоченное **комьюнити**, огромное количество библиотек:
 - + Около 4000 библиотек в CRAN.
 - Ядро содержит только самую основную функциональность, для многих часто возникающих задач уже нужно устанавливать библиотеки.
- + Во многом чрезвычайно **элегантный и интуитивный синтаксис**.
 - + Код читаем ⇒ анализ данных **воспроизводим!**
- + Широкие возможности для работы с **графикой**.

Декларативный бесплатный язык программирования. Появился в 1993 году. Это потрясающе гибкое приложение и язык для исследования, визуализации и понимания данных. На сегодняшний день является стандартом в области анализа данных.

Options for R on Hadoop

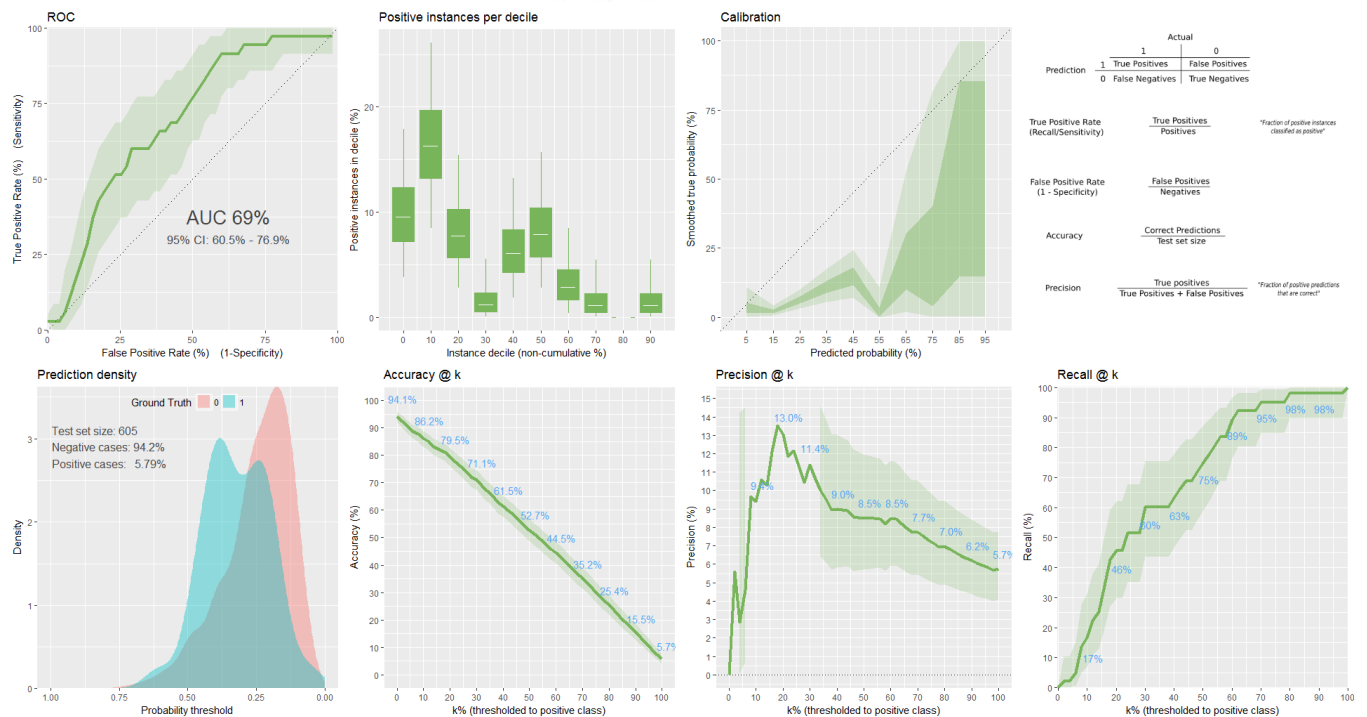
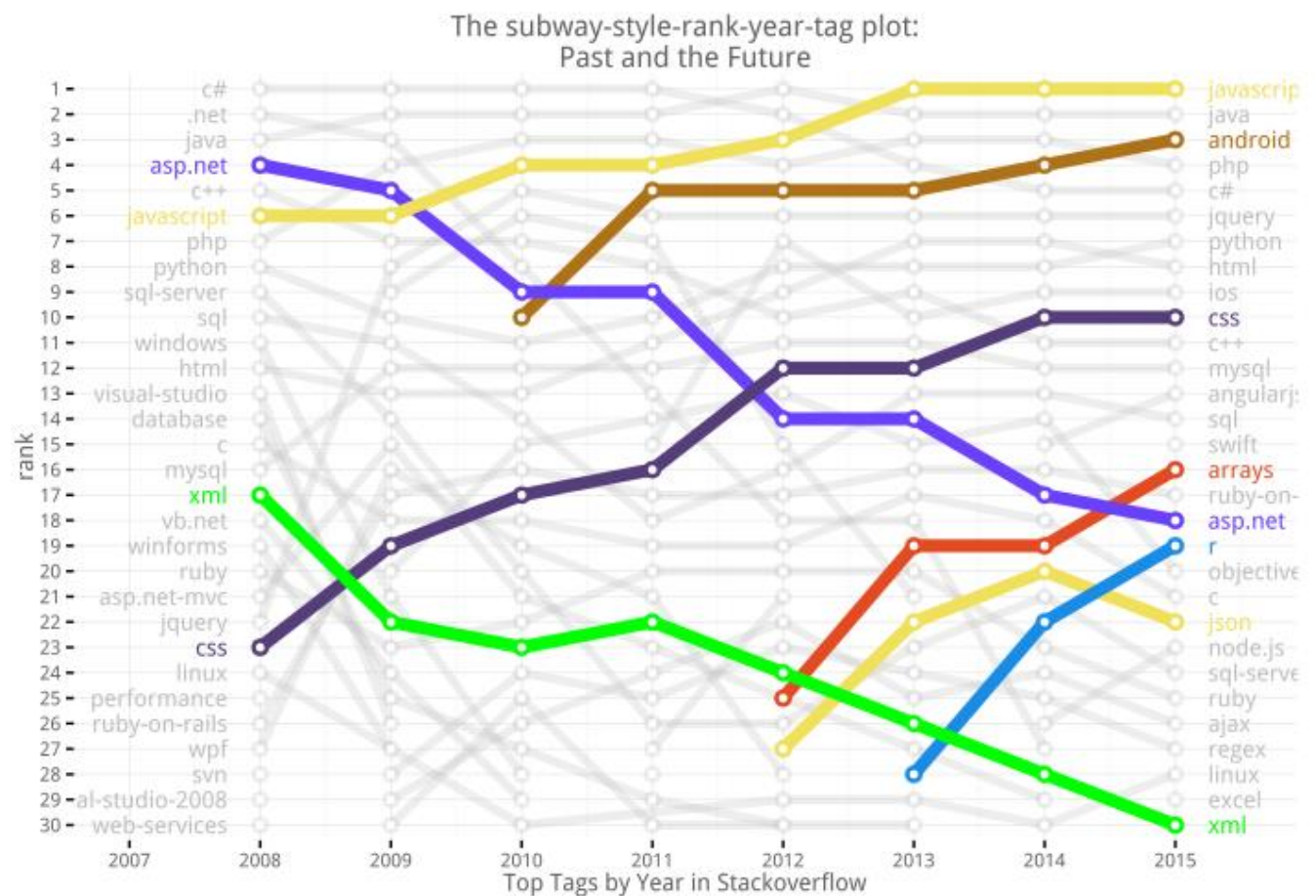
	RODBC/RJDBC	R Hive	RHadoop
Focus	<ul style="list-style-type: none"> SQL Access from R 	<ul style="list-style-type: none"> Broad access to Hive and HDFS 	<ul style="list-style-type: none"> Tight integration with core Hadoop components
Integration Ease	<ul style="list-style-type: none"> Install Hortonworks Hive ODBC driver Install Hive Libraries 	<ul style="list-style-type: none"> Requires Hadoop binaries, libraries, and configuration files on client machines Uses Java DFS Client and HiveServer 	
Benefits	<ul style="list-style-type: none"> Low impact on existing R scripts leveraging other DB packages Not required to install Hadoop configuration/binaries on client machines 	<ul style="list-style-type: none"> Wide range of features expressed through HQL - <i>hive-apply R Distributed apply function using HQL</i> 	<ul style="list-style-type: none"> Ability to run R on a massively distributed system Ability to work with full data sets instead of sample sets
Limitations	<ul style="list-style-type: none"> Parallelism limited to Hive Result set size 	<ul style="list-style-type: none"> Requires heavy client deployment Dependent on HiveServer, and can't be used with HiveServer2 	<p>Additional Information</p> <p>https://github.com/RevolutionAnalytics/RHadoop/wiki</p>

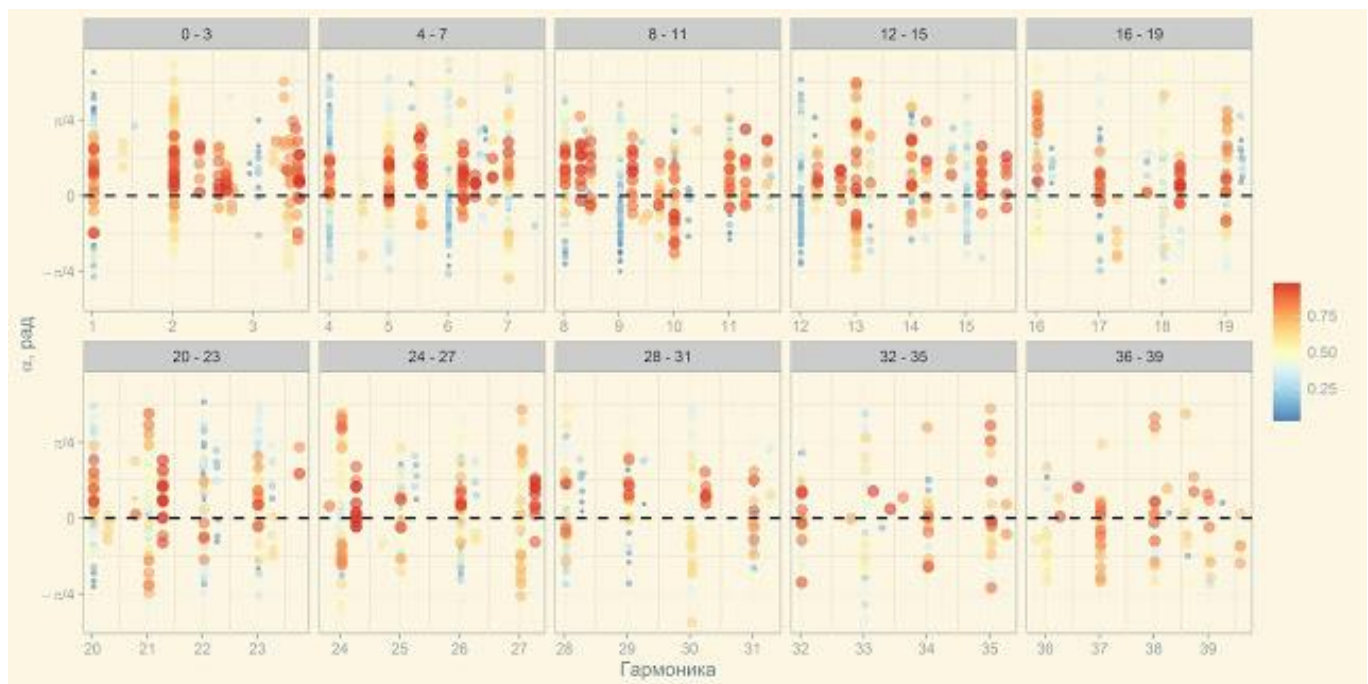
© Copyright 2014 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. HP No part is to be reproduced without written permission from HP.



R «заточен» под статистическую обработку данных, работу с графикой и алгоритмами машинного обучения. R поддерживает широкий спектр статистических и численных методов и обладает хорошей расширяемостью с помощью пакетов. Пакеты представляют собой библиотеки для работы специфических функций или специальных областей применения (например, доступ к BigData, у Oracle есть библиотека для работы с ее RDBMS, парсинга web-страниц и так далее). Используется единый глобальный репозиторий, хотя можно установить новые пакеты из любого источника.

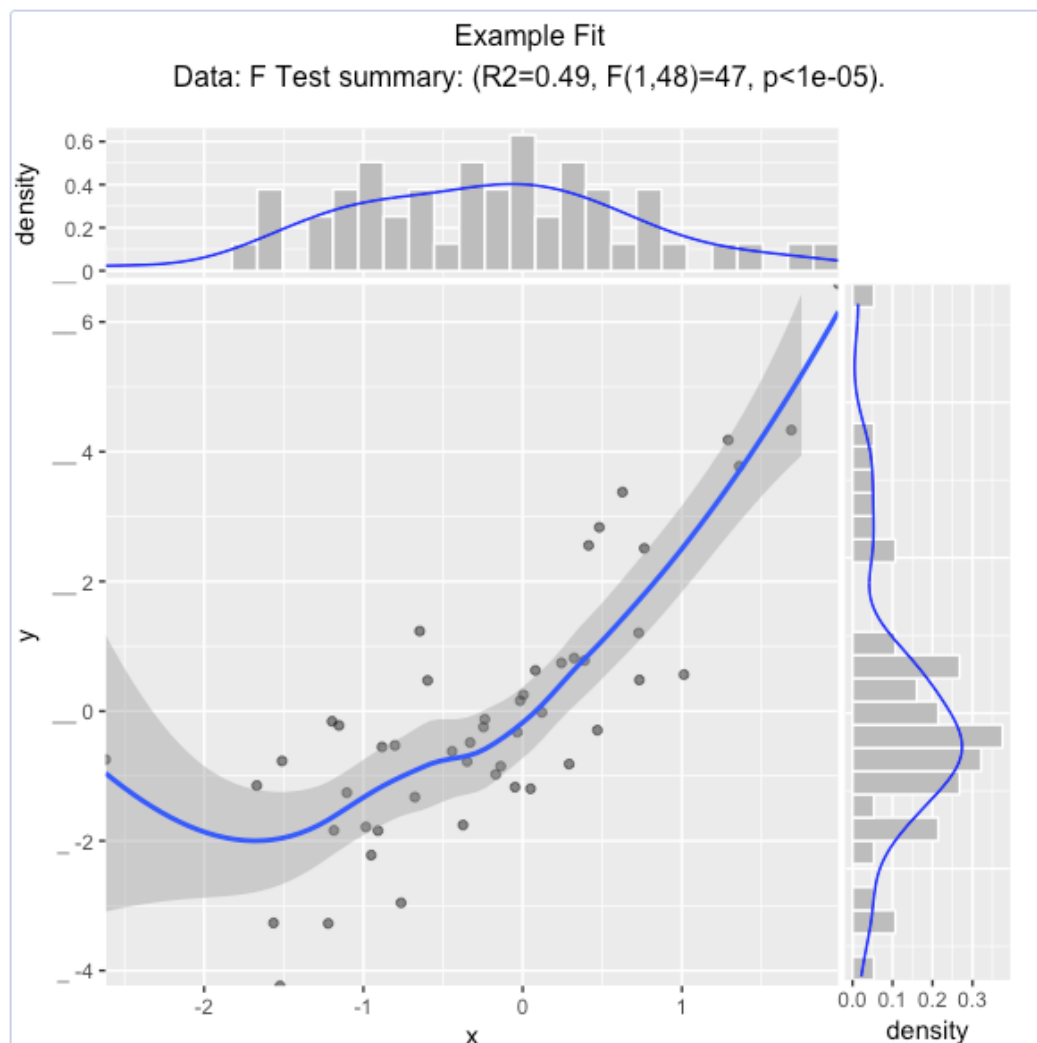
Ещё одной особенностью R являются графические возможности, заключающиеся в возможности создания качественной графики, которая может включать математические символы, в том числе интерактивные 3D-модели на JavaScript.





```
WVPlots::ScatterHist(frm, "x", "y", title="Example Fit")
```

```
## `geom_smooth()` using method = 'loess'
## `geom_smooth()` using method = 'loess'
```




Scatterplot with best linear fit through points. Reports the R-squared and significance of the linear fit.

2. Объекты языка R

2.1. Простые Numeric, Complex, Date, Character, Boolean, Raw

2.2. Векторы, Списки, Массивы, Матрицы, Факторы и, пожалуй, самый важные – DataFrame (чем-то похожа на таблицы в реляционных базах)

2.2.1. Data Frame - создает таблицу данных из поименованных или непоименованных аргументов

```
Console ~/ 
> a <- data.frame(Number=1:4, Letter=c("a","b","c","d"), Boolean=c(FALSE,TRUE,TRUE,FALSE))
> a
  Number Letter Boolean
1      1      a  FALSE
2      2      b   TRUE
3      3      c   TRUE
4      4      d  FALSE
> a$Number
[1] 1 2 3 4
> a[["Number"]]
[1] 1 2 3 4
> a[1]
  Number
1      1
2      2
3      3
4      4
> a[-1]
  Letter Boolean
1      a  FALSE
2      b   TRUE
3      c   TRUE
4      d  FALSE
> a[2:3]
  Letter Boolean
1      a  FALSE
2      b   TRUE
3      c   TRUE
4      d  FALSE
> a[c("Letter","Boolean")]
  Letter Boolean
1      a  FALSE
2      b   TRUE
3      c   TRUE
4      d  FALSE
> a[1,]
  Number Letter Boolean
1      1      a  FALSE
> a[,1]
[1] 1 2 3 4
> a[2,2]
[1] b
Levels: a b c d
```

2.3. Зарезервированные

2.3.1. NULL - "пустой" объект

2.3.2. NaN - неверная операция, например деление на 0


2.3.3. Inf - бесконечность (бывает как со знаком «плюс», так и со знаком «минус»)

2.3.4. NA - неизвестное значение (например, когда увеличили длину вектора без его заполнения)

3. Команды языка R


3.1. Полезная функция `help(...)` – получить документацию по указанной команде/функции

3.2. Присвоение

```
Console ~/   
> a <- c(1,2,3)  
> a  
[1] 1 2 3  
> a = seq(from = 1, to = 3, by = 1)  
> a  
[1] 1 2 3  
> c(1:3) -> a  
> a  
[1] 1 2 3  
> length(a)<-6  
> a  
[1] 1 2 3 NA NA NA  
> |
```


3.2.1. Интересное использование присвоения

3.3. Арифметические


```
Console ~/   
> a+1  
[1] 2 3 4 NA NA NA  
> b <- c(0.1,0.2,0.3)  
> a * b  
[1] 0.1 0.4 0.9 NA NA NA  
> |
```

3.4. Условные операторы if(), ifelse()

3.5. Циклы for, while, бесконечный цикл repeat

```
Console ~/ 
> for (k in 1:5){ print(k) }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
> k <- 1; while (k <= 5){ print(k); k<-k+1 }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
> k <- 1; repeat { print(k); k<-k+1; if(k>5) break }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
> |
```

3.6. Функции. Объявляются с помощью ключевого слова function

```
Console ~/ 
> fn=function(a) {
+   if (a) {
+     print("true");
+   } else {
+     print("false");
+   }
+ }
> fn(TRUE)
[1] "true"
> fn(F)
[1] "false"
> |
```

4. Пример расчета прогноза продаж и сравнение с данными из Oracle Demantra

4.1. Введение. У ритейла есть большое количество торговых точек, где продаются разные продукты: сотовые телефоны, аксессуары и другое. На основании данных продаж в Oracle Demantra строится прогноз на ближайшие 6 недель о том, сколько будет еще продано. Из-за объемов и "тормозов" самой Demantra'y делаются упрощения: данные берутся не за каждый день, а за всю неделю, а также не по каждому элементу, а по группе элементов (типа, смартфоны iPhone 6s 64Гб, без учета цвета), и нет различий по точкам продаж. В нашем примере сделаем несколько упрощений (учитываем только продажи, и не учитываем дефицит и остатки; не учитываем «выбросы»; только простая формула регрессии – линейная; другие параметры). Так, возьмём только одну группу - "Роутер_4G_WIFI_MAIN" с кодом 12225. На этом примере увидим также, как использовать конструкции в языке R.

4.2. Настроим среду

```
Console C:/Users/andrey.zvyagin.DIR/Downloads/Smart Tip/Moe/
> Sys.setenv(JAVA_HOME='C:\\Program Files (x86)\\Java\\jre1.8.0_141')
> install.packages("rJava")
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/rJava_0.9-8.zip'
Content type 'application/zip' length 716884 bytes (700 KB)
downloaded 700 KB

package 'rJava' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:\Users\andrey.zvyagin.DIR\AppData\Local\Temp\RtmpCSfk1Q\downloaded_packages
> # добавим работу с Excel
> install.packages("xlsx", dep = T)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/xlsx_0.5.7.zip'
Content type 'application/zip' length 401348 bytes (391 KB)
downloaded 391 KB

package 'xlsx' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:\Users\andrey.zvyagin.DIR\AppData\Local\Temp\RtmpCSfk1Q\downloaded_packages
> library("xlsx")
Загрузка требуемого пакета: rJava
Загрузка требуемого пакета: xlsxjars
> library(ggplot2)
> library(dplyr)

Присоединяю пакет: 'dplyr'

Следующие объекты скрыты от 'package:lubridate':

  intersect, setdiff, union

Следующие объекты скрыты от 'package:xts':

  first, last

Следующие объекты скрыты от 'package:stats':

  filter, lag

Следующие объекты скрыты от 'package:base':

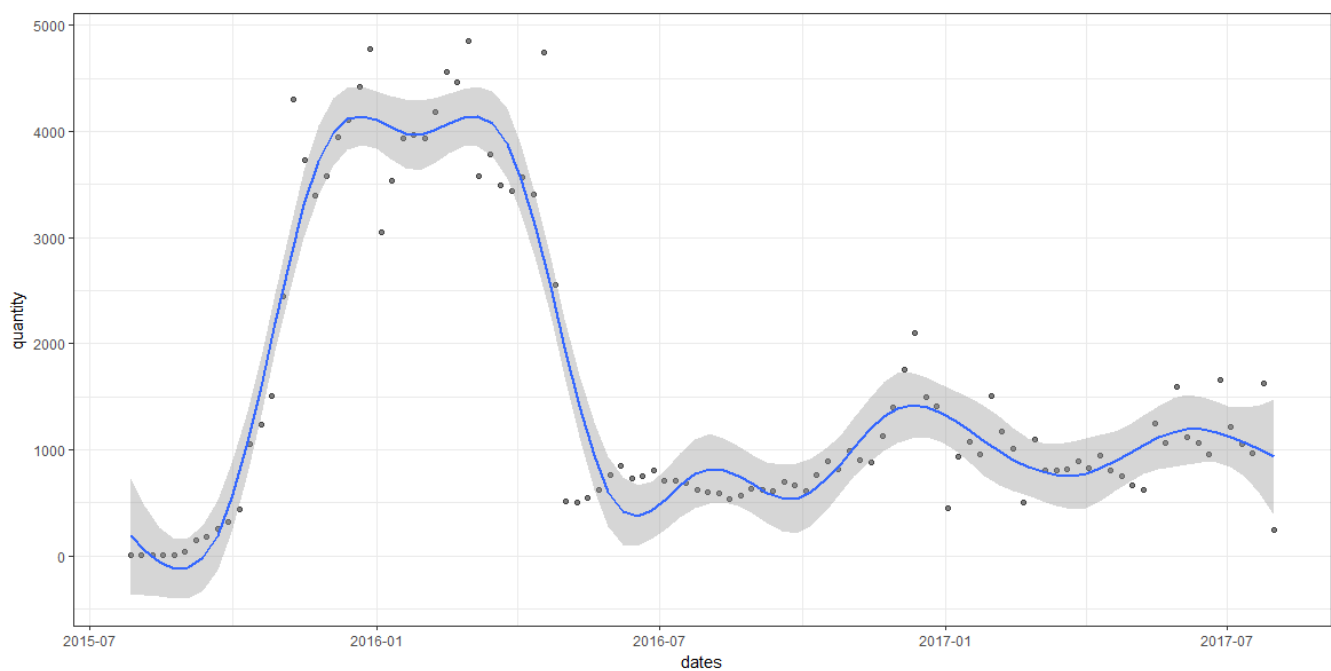
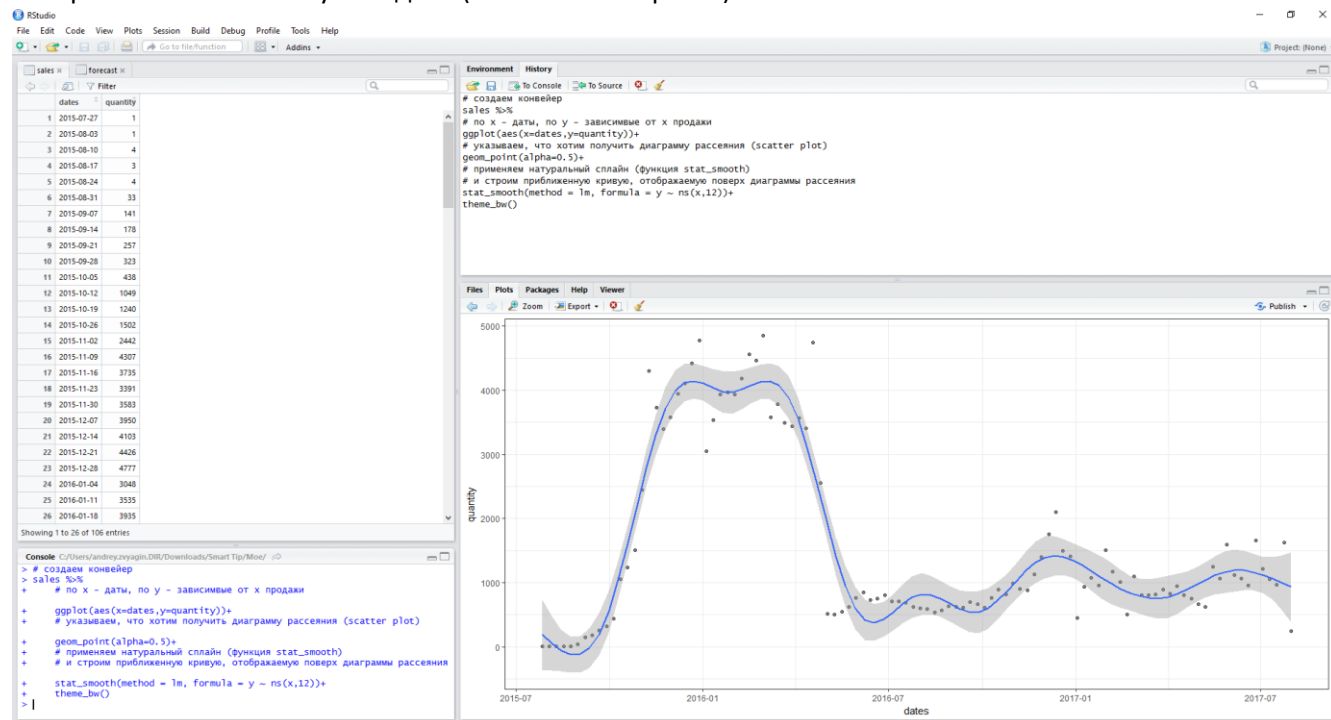
  intersect, setdiff, setequal, union

> |
```

4.3. Загрузим исходные данные по продажам

```
Console C:/Users/andrey.zvyagin.DIR/Downloads/Smart Tip/Moe/
> # загрузим данные по продажам
> setwd("C:/Users/andrey.zvyagin.DIR/Downloads/Smart Tip/Moe")
> sales1 <- read.xlsx("sales.xls", sheetIndex = 1); sales2 <- read.xlsx("sales.xls", sheetIndex = 2); sales3 <- read.xlsx("sales.xls", sheetIndex = 3)
> # подготовим данные
> dates1<-sales1$SALEDATE[rep(which(sales1$DIST_MOD_ID==12225))]; dates2<-sales2$SALEDATE[rep(which(sales2$DIST_MOD_ID==12225))]; dates3<-sales3$SALEDATE[rep(which(sales3$DIST_MOD_ID==12225))];
> dates<-c(dates1,dates2,dates3)
> quantity1<-sales1$QUANTITY[rep(which(sales1$DIST_MOD_ID==12225))]; quantity2<-sales2$QUANTITY[rep(which(sales2$DIST_MOD_ID==12225))]; quantity3<-sales3$QUANTITY[rep(which(sales3$DIST_MOD_ID==12225))]
> quantity<-c(quantity1,quantity2,quantity3)
> quantity<-quantity[order(dates)] # сортировка
> dates<-dates[order(dates)] # сортировка
> sales<-data.frame(dates,quantity)
> head(sales)
  dates quantity
1 2015-07-27      1
2 2015-08-03      1
3 2015-08-10      4
4 2015-08-17      3
5 2015-08-24      4
6 2015-08-31     33
> head(sales1)
  DIST_MOD_ID SALEDATE QUANTITY
1         29571 2016-01-25       36
2         11131 2016-08-08       42
3         10344 2017-07-03       18
4         30283 2017-03-27        3
5         10314 2015-11-30        3
6         10380 2016-12-05        2
> |
```

4.4. Построим математическую модель (Natural Cubic Splines)



4.5. Загрузим данные по прогнозу из Oracle Demantra на ближайшие 6 недель

```

Console C:/Users/andrey.zvyagin.DIR/Downloads/Smart Tip/Moe/
> # добавим на график данные из Oracle Demantra
> forecast <- read.xlsx("forecast.xls", sheetIndex = 1)
> forecast_s<-forecast$SALEDATE[rep(which(forecast$DIST_MOD_ID==12225))]
> forecast_q<-forecast$QUANTITY[rep(which(forecast$DIST_MOD_ID==12225))]
> forecast_q<-forecast_q[order(forecast_s)] # сортировка
> forecast_s<-forecast_s[order(forecast_s)] # сортировка
> forecast<-data.frame(dates=forecast_s,quantity=forecast_q)
> forecast
  dates quantity
1 2017-07-31    672
2 2017-08-07    738
3 2017-08-14    672
4 2017-08-21    680
5 2017-08-28    718
6 2017-09-04    639
>

```

4.6. Построим сами прогноз продаж на ближайшие 6 недель

```

Console C:/Users/andrey.zvyagin.DIR/Downloads/Smart Tip/Moe/
> # вычислим прогноз
> # 1. построим линейную регрессионную модель
> fit.lm<-lm(quantity~ns(dates,12),data=sales)
> summary(fit.lm)

Call:
lm(formula = quantity ~ ns(dates, 12), data = sales)

Residuals:
    Min       1Q   Median       3Q      Max
-1391.42  -186.26    8.68   159.34  1985.38

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)      185.9       274.0   0.679  0.499043
ns(dates, 12)1    4799.8       369.6  12.987 < 2e-16 ***
ns(dates, 12)2    3268.4       462.9   7.061 2.95e-10 ***
ns(dates, 12)3    4794.8       417.8  11.477 < 2e-16 ***
ns(dates, 12)4    -837.1       443.1  -1.889 0.061989 .
ns(dates, 12)5    1178.7       429.9   2.742 0.007332 **
ns(dates, 12)6    -166.6       437.0  -0.381 0.703949
ns(dates, 12)7    1615.5       433.0   3.731 0.000328 ***
ns(dates, 12)8     802.4       434.3   1.848 0.067824 .
ns(dates, 12)9     375.3       430.4   0.872 0.385511
ns(dates, 12)10   1462.8       360.8   4.054 0.000104 ***
ns(dates, 12)11    134.0       702.2   0.191 0.849040
ns(dates, 12)12  1257.7       315.6   3.985 0.000134 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 447.9 on 93 degrees of freedom
Multiple R-squared:  0.9085,    Adjusted R-squared:  0.8967
F-statistic: 76.96 on 12 and 93 DF,  p-value: < 2.2e-16

> # 2. сам прогноз
> myforecast<-data.frame(dates=forecast$dates)
> mypredict<-predict(fit.lm,myforecast)
> mypredict
      1      2      3      4      5      6
932.7451 880.1100 827.4749 774.8398 722.2047 669.5696
>

```

	dates	quantity
1	2017-07-31	672
2	2017-08-07	738
3	2017-08-14	672
4	2017-08-21	680
5	2017-08-28	718
6	2017-09-04	639

Showing 1 to 6 of 6 entries

4.7. Сравним на графике наши результаты

4.7.1. Отличия в расчетах есть, так как в Demantra используется несколько методов: убирание всплесков, разные модели регрессии и выбор из них лучшей и так далее. В рамках этого смарт-типа это не рассматривается.

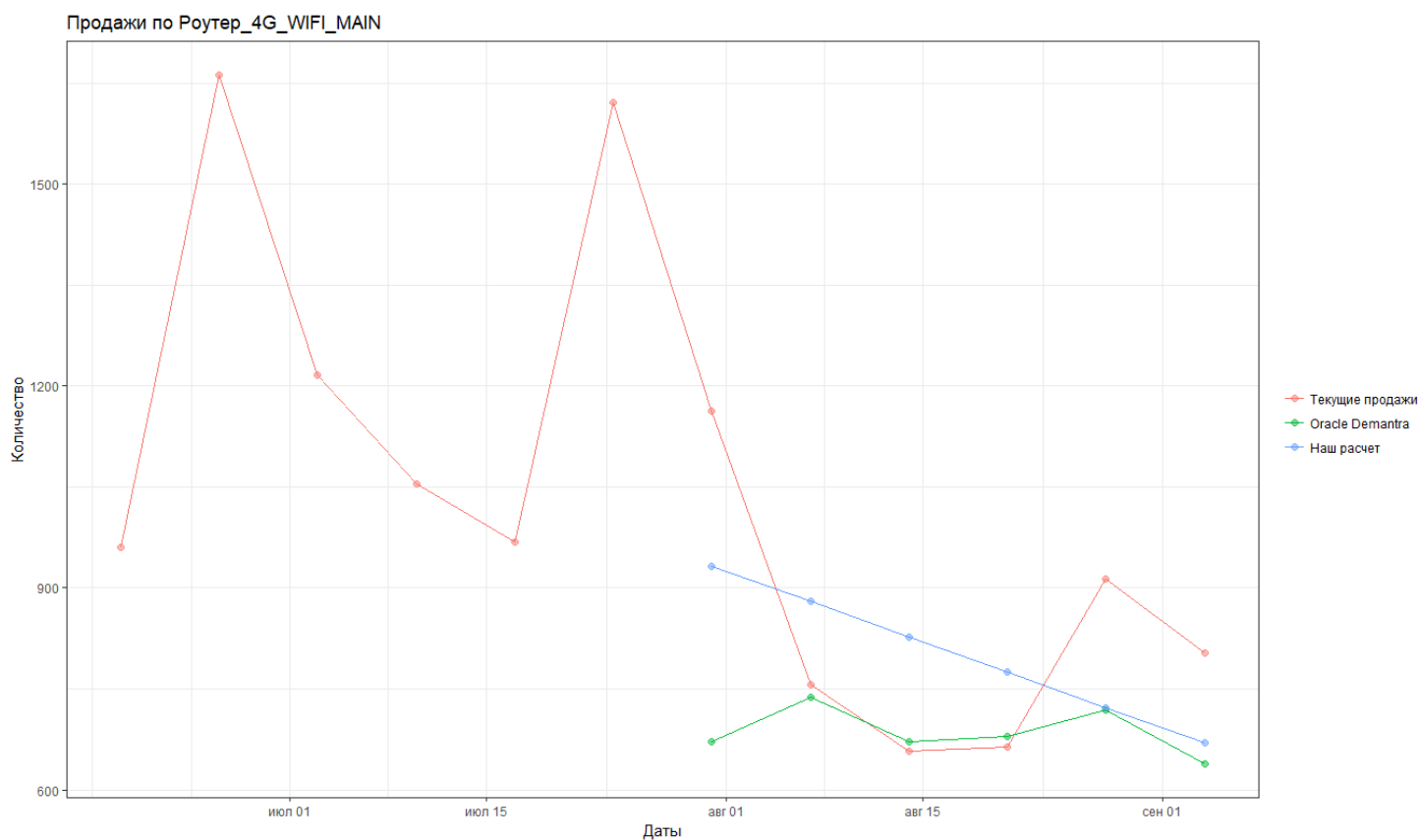
























График Oracle Demantra очень сильно пересекается с графиком реальных продаж из-за того, что реальные остатки долгое время (несколько недель) были «прижаты» тем, сколько было заказано ритейлом на основании прогноза из Oracle Demantra.

5. Перспективы языка R

5.1. Выводы:

- очень удобная работа с данными: массивами и так далее (минимум запросов и кода)
- прекрасные возможности по визуализации данных, инфографика
- очень просто производить очень сложные вычисления (построение запроса буквально в 2 строки)
- большое количество библиотек для большого круга задач: статистика и прочие научные работы, машинное обучение и другое. Новые функции становятся доступными для скачивания еженедельно
- можно не только делать сложные вычисления, но и увидеть логику данных вычислений
- интерактивность (можешь сразу же проверить какую-то теорию, увидеть результаты)
- хорошо документирован
- есть интерфейс для внешних библиотек на других языках программирования
- доступны графические интерфейсы пользователя, которые позволяют строить модели по типу drag-and-drop (Weka и другие)
- бесплатность

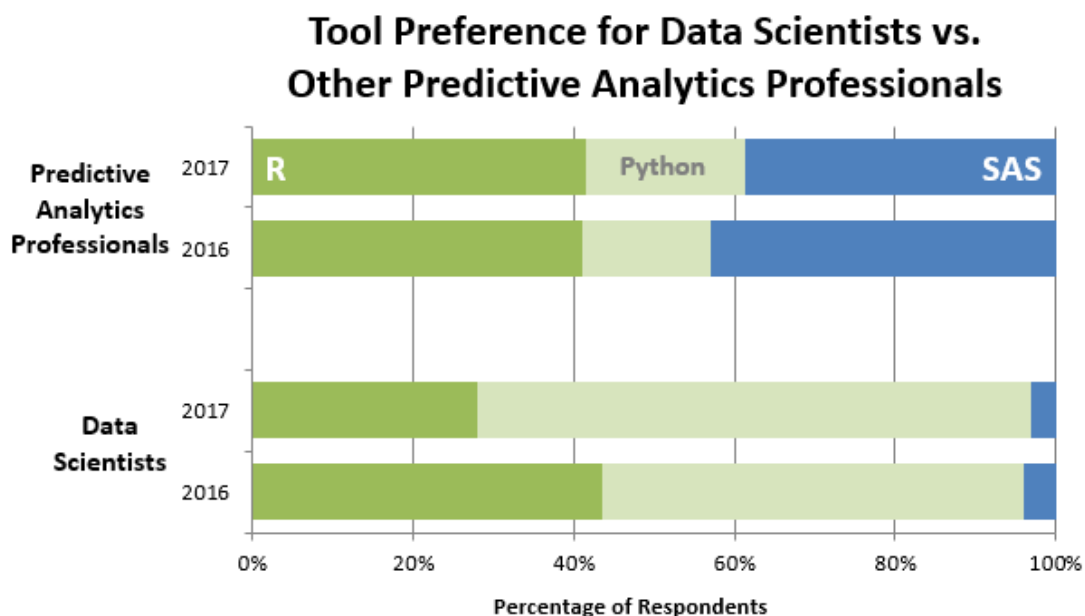
The 2017 Top Programming Languages

Language Rank	Types	Spectrum Ranking
1. Python	 	100.0
2. C	  	99.7
3. Java	  	99.5
4. C++	  	97.1
5. C#	  	87.7
6. R		87.7
7. JavaScript	 	85.6
8. PHP		81.2
9. Go	 	75.1
10. Swift	 	73.7

[The 2017 Top Programming Languages \(IEEE Spectrum\)](#)

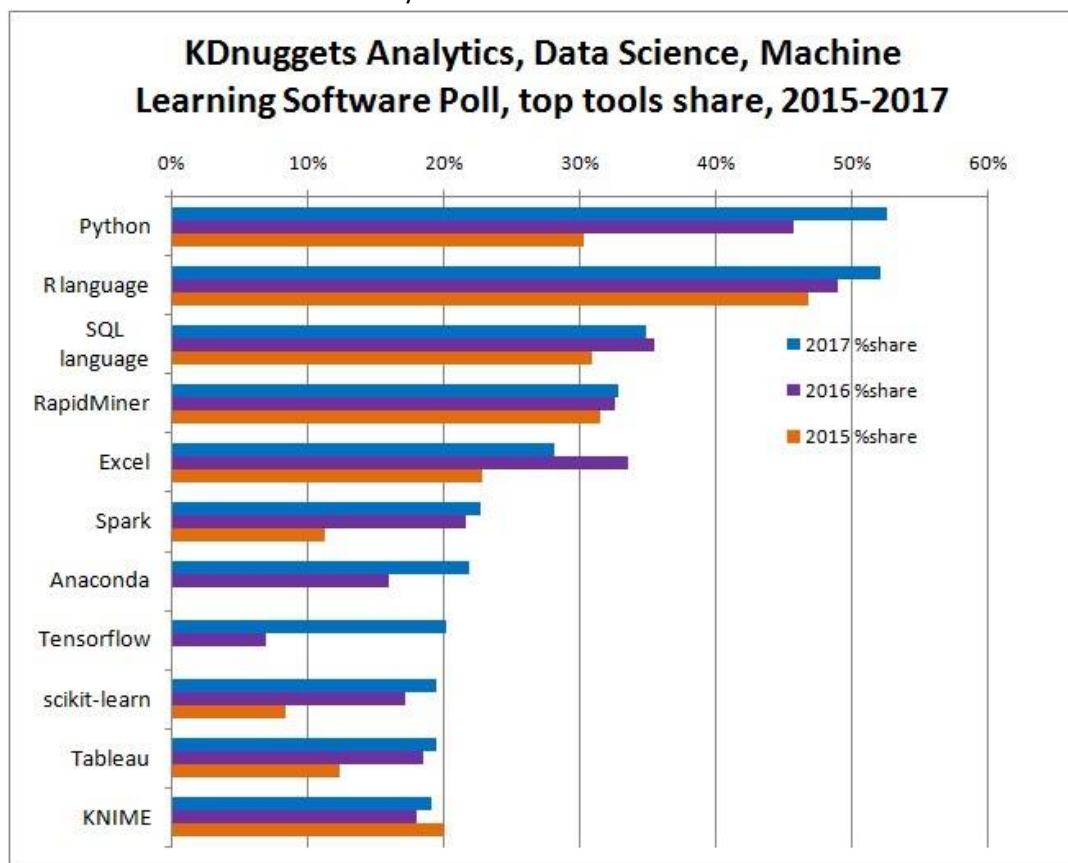
5.2. Основной конкурент - язык Python

- 5.2.1. R проигрывает Python: Python - это универсальный язык, более распространен, R больше для интерактивной работы
- 5.2.2. R выигрывает Python: больше библиотек, один глобальный репозиторий пакетов и в целом система их установки, скорость R (все библиотеки на C/C++), многовариантность решения задач, простота в работе, больше зарплата
- 5.2.3. SAS можно не рассматривать, так как он используется в очень крупном энтерпрайзе



[2017 SAS, R, or Python Flash Survey Results \(Burtch Works Executive Recruiting\)](#)

5.2.4. Очень часто аналитики используют оба языка



[New Leader, Trends, and Surprises in Analytics, Data Science, Machine Learning Software Poll \(KDnuggets News\)](#)

6. Ссылки

- 6.1. [Основной сайт проекта](#)
- 6.2. [CRAN \(Comprehensive R Archive Network\) — центральная система хранения и распространения R и его пакетов](#)
- 6.3. [Язык программирования R — Викиучебник](#)
- 6.4. [Блог "R: Анализ и визуализация данных"](#)
- 6.5. [Python & R codes for common Machine Learning Algorithms](#)