

Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Филологический факультет

УТВЕРЖДЕНО:

Декан

И. В.Тубалова

Рабочая программа дисциплины

Информатика и основы программирования

по направлению подготовки

45.03.03 Фундаментальная и прикладная лингвистика

Направленность (профиль) подготовки:

Фундаментальная и прикладная лингвистика

Форма обучения

Очная

Квалификация

Бакалавр

Год приема

2024

СОГЛАСОВАНО:

Руководитель ОП

А.В. Васильева

Председатель УМК

Ю.А. Тихомирова

Томск – 2024

1. Цель и планируемые результаты освоения дисциплины

Целью освоения дисциплины является формирование следующих компетенций:

- ПК-4 – разрабатывать программный код при решении задач автоматической обработки текстов;
- ОПК-7 – Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности

Результатами освоения дисциплины являются следующие индикаторы достижения компетенций:

ИПК-4.1 Применяет способы формализации и алгоритмизации поставленных задач в сфере автоматической обработки текстов.

ИПК-4.2 Создает программный код с использованием языков программирования и манипулирования данными в сфере автоматической обработки текстов.

ИОПК-7.3 Применяет современные технические средства и информационно-коммуникационные технологии для решения задач профессиональной деятельности.

2. Задачи освоения дисциплины

- 1) формирование понимания студентами ключевых положений информатики, ее структуры, связи с другими науками, в том числе с лингвистикой (на основе языка программирования R), объяснение базовых правил разработки алгоритмов и программ;
- 2) приобретение студентами уверенных навыков практической работы (программирования) на компьютере, формирование основ для последующих курсов, посвященных созданию современных информационных систем и разработке программ;
- 3) ознакомление студентов с парадигмами (стилями) программирования и механизмами их реализации в программах – основной акцент делается на процедурном программировании и формировании «аккуратного» и постепенного перехода к обсуждению начальных положений программирования на примере лингвистических данных;
- 4) ознакомление студентов с основными алгоритмами и программами в сфере обработки естественного языка.

3. Место дисциплины в структуре образовательной программы

Дисциплина относится к обязательной части образовательной программы.

4. Семестр(ы) освоения и форма(ы) промежуточной аттестации по дисциплине

Первый семестр, зачет

Второй семестр, зачет с оценкой

5. Входные требования для освоения дисциплины

Для успешного освоения дисциплины требуются компетенции, сформированные в ходе освоения образовательных программ предшествующего уровня образования.

6. Язык реализации

Русский

7. Объем дисциплины

Общая трудоемкость дисциплины составляет 4 з.е., 144 часов, из которых:

-лекции: 16 ч.

-практические занятия: 48 ч.

в том числе практическая подготовка: 48 ч.

Объем самостоятельной работы студента определен учебным планом.

8. Содержание дисциплины, структурированное по темам

Тема 1. Введение в программирование

Типы данных и работа с ними. Вектор, матрица, массив

Тема 2. Работа с текстовым массивом данных

Тема 3. Статистический анализ

Тема 4 Визуализация данных

Тема 5. Автоматический морфологический анализ.

Тема 6. Машинное обучение: без учителя. Метрические алгоритмы классификации и кластеризации

Тема 7. Методы сокращения выборки

Тема 8. Машинное обучение с учителем.

Тема 9. Метод опорных векторов (SVM)

Тема 10. Решающие списки и деревья

Тема 11. Оценка результатов машинного обучения (нормальные метрики оценки модели: accuracy, precision, recall, f1-score)

Тема 12. Нейронные сети. Типологизация НС. Генерация текстов с помощью НС.

9. Текущий контроль по дисциплине

Текущий контроль образовательной программы (темы, раздела, модуля) требованиям образовательных стандартов по направлениям подготовки/специальностям. Текущий контроль успеваемости обучающихся направлен на определение соответствия результатов обучения после освоения элемента по дисциплине проводится путем контроля посещаемости, проведения контрольных работ, разработки кода, выполнения домашних заданий, практических задач, направленных на формирование понимания основ программирования (переменные, циклы, функции, базовые алгоритмы - ОПК-7); применение на практике существующих алгоритмов обработки естественного языка (ПК-4 и ОПК-7), написание программного кода, применение и понимание методов машинного обучения, нейронных сетей (ИПК-4.1, ИПК-4.2, ИОПК-7.3). Результаты освоения компетенций фиксируются в форме контрольной точки не менее одного раза в семестр.

При фиксации текущего контроля в курсе «Moodle» применяется 100-балльная система. Оценка за текущий контроль соответствует международному принципу ранжирования Болонской системы оценивания сформированных компетенций и переводится в традиционную пятибалльную. Система перевода 100-балльной оценки в пятибалльную:

90-100 баллов – «отлично» (или А. Пороги: 90-94 – 5- (А-); 95-100 – 5 (А)),

76-89 баллов – «хорошо» (или В. Пороги: 76-80 – 4- (В-); 81-89 – 4 (В)),

56-75 баллов – «удовлетворительно» (или С. Пороги: 56-65 – 3- (С-); 66-75 – 3 (С)),

Менее 55 баллов – «неудовлетворительно (или D)»

10. Порядок проведения и критерии оценивания промежуточной аттестации

Зачет в первом семестре проводится в виде решения практического задания, который предусматривает развитие следующих компетенций: ПК-4, ИПК-4.1, ИПК-4.2. И направлен на формирование базовых навыков программирования, сбора, структуризации текстовых данных и их первичной обработки (предобработка).

Примеры задач:

1. Создайте парсер для новостного агентства:

Пример кода:

```
page_start <- "https://news.vtomske.ru/c/tomsk"
page_full <- paste0(page_start, pages2)
vec_links <- c()
i <- 1
page_start <- "https://news.vtomske.ru/c/tomsk"
part_pages <- "?down=1637120100"
page_links <- c()
for (i in 1:139) {
  page_full <- paste0(page_start, part_pages)
  webpage <- read_html(page_full)
  part_pages <- webpage %>% html_nodes(".prev") %>%
    html_attr("href")
  page_links[i] <- paste0(page_start, part_pages)
  #page_links[i] <- page_full
  i=i+1
}
```

Дифференцированный зачет по изучаемой дисциплине основан на проектном подходе и представляет собой презентацию проекта, разработку кода с учетом выбранного массива текстов, применения алгоритмов предобработки текстов и методов машинного обучения в задачах обработки естественного языка (ПК-4, ОПК-7, ИПК-4.1, ИПК-4.2, ИОПК-7.3). Результаты работы алгоритмов подтверждаются на основании формальных метрик оценивания.

Примерные задания по 5 модулю:

Изучите код, инициализирующего лемматизацию массива текстов. Примените код на своих текстах, которые были скачаны в прошлом семестре

```
data_path = "D:/stem"
list.files(path = "D:/stem", "txt")
# all subjects
#subject_id = c(c(1:3))
subject_id = list.files(path = "D:/stem", "txt")
# stem_id = c(c(1:3))
# write_id = c(c(1:3))
#filename = paste0(data_path, "/", subject_id[2])
for ( id in subject_id ) {
  #filename = paste0(data_path, "/dlg_m_", id, ".txt")
  filename = paste0(data_path, "/", id)
  print(filename)
  text <- read.table(filename, header = FALSE, sep = "\t", encoding =
"russian",quote="",comment="")
  ldf <- lapply(as.matrix(text$V1), function(x)x)
  text2 <- as.character(ldf)
```

```

#text <- as.character(text$V1)
text3 <- enc2utf8(text2)
cmd <- "D:/stem/mystem.exe -cnild -e UTF-8"
#if (!is.null(fixlist) && file.exists(fixlist))
# cmd <- paste(filename, + id)
#system(cmd, intern = TRUE, input = read.table(filename, header = TRUE, sep = "\t",
encoding = "UTF-8"))
# stem <- system(cmd, intern = TRUE, input = as.character(read.table(filename, header =
TRUE, sep = "\t", encoding = "UTF-8")), output = write(stem,"0_"))
stem <- system(cmd, intern = TRUE, input = text3)
#for(id in subject_id){
myfile<-paste0("0_", id)
# write.table(stem,myfile,sep="\t",row.names=FALSE, col.names=F,)
#}
write.table(stem, myfile, row.names=F,col.names=F,sep="\t")
}

```

Пример задания темы 8

```
#####Bayes classifier#####
```

```
# Loading package
```

```
library(e1071)
```

```
library(caTools)
```

```
library(caret)
```

```
# Fitting Naive Bayes Model
```

```
# to training dataset
```

```
classifier_auth <- naiveBayes(Author ~ ., data = author.numtr)
```

```
classifier_auth
```

```
# Predicting on test data'
```

```
y_pred <- predict(classifier_auth, newdata = author.numte)
```

```
# Confusion Matrix
```

```
cm <- table(author.numte$Author, y_pred)
```

```
cm
```

Примените алгоритм классификации Байеса на своих данных (частотной матрицы).

Критерии зачета обусловлены логической демонстрацией приобретенных компетенций в соответствии с текущей программой. Демонстрация предусматривает уверенное использование терминологии, понимание и корректное использование математического аппарата, предусматривает корректность написания кода, его понимание и корректное использование в нем математических методов. Отметка «зачтено» выставляется за счет демонстрации полученных компетенций в практиках, домашних работах и итоговом задании: уверенное владение и понимание работы кода, знание и демонстрация в практике теоретических основ баз данных. Минимальный порог зачета составляет 55 баллов, ниже 55 – «не зачтено»

Дифференцированный зачет во втором семестре также предусматривает ответы на теоретические вопросы в устной форме по билетам

Ответы на вопросы предполагают решение задач, краткую интерпретацию полученных результатов и проверяют уровень сформированности следующих компетенций: ОПК-7, ИОПК-7.3.

Примерный перечень теоретических вопросов

1. Опишите понятие энтропии Шеннона, дайте примеры использования. В каком методе машинного обучения используется энтропия?
2. Чем отличаются цели классификации и регрессии в машинном обучении?
3. Какие методы векторизации используются в репрезентации текстового массива данных? Опишите преимущества и недостатки методов
4. В каких случаях применяется индекс прироста информации? Опишите алгоритм его работы.
5. Опишите формальные метрики точности работы классификаторов. В чем преимущества и недостатки формальных метрик?

Примеры практических задач:

1. Задача 1. Построение линейных классификаторов

Дано: Матрица для обучения с признаками, исходный код двух классификаторов:

```
##LDA
author.lda<-lda(Author~.,data=author.numtr)
print(author.lda$scaling)
summary(author.lda)
author.ldapredicttrain<-predict(author.lda,author.numtr)
tldatrain<-table(author.numtr$Author,author.ldapredicttrain$class)
error(tldatrain)
f1(tldatrain)
author.ldapredicttest<-predict(author.lda,author.numte)
summary(author.ldapredicttest)
tldatest<-table(author.numte$Author,author.ldapredicttest$class)
error(tldatest)
print(paste0(c("text: ", 55)))
x1 <- "string1"
x2 <- "string2"
paste0(x1, x2)
f1(tldatest)
plot(author.lda,col=author.train$colour)
plot(author.lda,dimen=1,type="both")
plot(author.lda,dimen=1,type="density")
```

##Logistic Regression

```
lr<-multinom(Author~.,data=author.numtr)
help(multinom)
lr.train<-predict(lr,author.numtr,type = "class")
error(table(author.numtr$Author,lr.train))
```

```
lr.test<-predict(lr,author.numte,type = "class")
error(table(author.test$Author,lr.test))
```

##SVM

```
# author.numtr$colour = NA
# author.numtr$colour[author.numtr$Author == "Austen"] = 0
# author.numtr$colour[author.numtr$Author == "London"] = 1
# author.numtr$colour[author.numtr$Author == "Milton"] = 2
# author.numtr$colour[author.numtr$Author == "Shakespeare"] = 3
# author.numtr <- author.numtr[,-15]
author.numtr$Author <- as.factor(author.numtr$Author)
author.svm<-svm(Author~.,data=author.numtr,
kernel="sigmoid")
```

```
help("svm")
summary(author.svm)
```

```
author.pred<-predict(author.svm,
author.numtr,decision.values=T)
ttrain<-table(author.numtr$Author,author.pred)
error(ttrain)
```

```
author.predtestsvm<-predict(author.svm,author.numte,decision.values=T)
ttest<-table(author.numte$Author,author.predtestsvm)
error(ttest)
```

Требуется: подобрать оптимальные гиперпараметры модели, влияющих на работу классификатора

1. Задача 2. Энтропия Шеннона

Дано: Исходные классы в векторе:

```
train <- c(1:20)
x <- train
target = c(
0, 1, 1, 1,
1, 0, 0, 0,
0, 1, 1, 1,
1, 0, 0, 0,
0, 0, 0, 1
)
ones <- c()
i=1
j =1
for (i in 1:length(target)){
if (target[i]==1){
ones[j] = target[i]
i=i+1
j=j+1
}else{
i=i+1
}
}
zeros <- c()
i=1
j =1
for (i in 1:length(target)){
if (target[i]==0){
zeros[j] = target[i]
i=i+1
j=j+1
}else{
i=i+1
}
}
p_1 <- length(zeros)/length(target)
p_0 <- length(ones)/length(target)
p <- c(p_0, p_1)
```

```
S0 = sum((- (p * log2(p))))
```

```
target2 <- c(target[1:13])
```

```
ones <- c()
```

```
i=1
```

```
j =1
```

```
for (i in 1:length(target2)){
```

```
  if (target2[i]==1){
```

```
    ones[j] = target2[i]
```

```
    i=i+1
```

```
    j=j+1
```

```
  }else{
```

```
    i=i+1
```

```
  }
```

```
}
```

```
zeros <- c()
```

```
i=1
```

```
j =1
```

```
for (i in 1:length(target2)){
```

```
  if (target2[i]==0){
```

```
    zeros[j] = target2[i]
```

```
    i=i+1
```

```
    j=j+1
```

```
  }else{
```

```
    i=i+1
```

```
  }
```

```
}
```

```
p_1 <- length(zeros)/length(target2)
```

```
p_0 <- length(ones)/length(target2)
```

```
p <- c(p_0, p_1)
```

```
S1 = sum((- (p * log2(p))))
```

```
# Прирост информации
```

```
IG = S0 - sum(target)/length(x) * S1 - sum(target2)/length(x)
```

Задача: подобрать оптимальную энтропию Шеннона и прирост информации

Результаты дифференцированного зачета определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Критерии дифференцированного зачета обусловлены логической демонстрацией приобретенных компетенций в соответствии с текущей программой. Демонстрация предусматривает уверенное использование терминологии, понимание и корректное использование математического аппарата, предусматривает корректность написания кода, его понимание и корректное использование в нем математических методов. Отметка «хорошо» выставляется за счет демонстрации полученных компетенций, владение и понимание кода, теоретических аспектов его применения в практике работы с текстовыми массивами данных допускаются недочеты в понятийном аппарате математики. Отметка «удовлетворительно» позволяет допустить ошибки в разработке кода, но учитывает последовательную логику изложения структуры кода, его интерпретацию, связь теоретических аспектов лингвистики и математики, демонстрация понимания хода обработки текста. Минимальный порог оценки «отлично» составляет 90-100 баллов, хорошо 75-89, удовлетворительно «55-74» ниже 55 – «неудовлетворительно»

11. Учебно-методическое обеспечение

а) Электронный учебный курс по дисциплине в электронном университете «Moodle» - <https://moodle.tsu.ru/course/view.php?id=7431>

б) Оценочные материалы текущего контроля и промежуточной аттестации по дисциплине: <https://moodle.tsu.ru/mod/page/view.php?id=751611>

б) Оценочные материалы текущего контроля и промежуточной аттестации по дисциплине.

в) План семинарских / практических занятий по дисциплине.

Тема 1. Введение в программирование. Введение в язык R. Синтаксис и управляющие конструкции на языке в R Команды, операторы. Объекты, функции. Начало работы с данными в языке программирования R. Основы программирования в R. Управляющие структуры. Условные операторы, циклы, функции. Основы программирования в R. Управляющие структуры. Условные операторы, циклы, функции. Типы данных и работа с ними. Вектор, матрица, массив

Тема 2. Работа с текстовым массивом данных. Парсинг массива текстовых данных и их представление в структурированном виде. Морфологический анализ скаченного массива текстов: mystem, Udipe. Векторизация слов: ONE, bow, word2vec. Создание лингвистических словарей. Частотный анализ ЛЕ: абсолютная и относительная частоты, TF-IDF

Тема 3. Статистический анализ. Основные типы распределений. Корреляции и регрессии. Корреляции Пирсона, Фишера. Линейная регрессия, логистическая регрессия. Описательные методы статистики. Статистические критерии проверки значимости. Самостоятельная работа по теории

Тема 4 Визуализация данных. Гистограмма, boxplot, dotchart, pie chart. Работа с библиотекой ggplot2. Вывод результатов в RMarkdown

Тема 5. Автоматический морфологический анализ. Типы морфологических анализаторов. Интеграция морфологического анализатора Яндекса «Mystem» в язык программирования R. Создание колонки лемматизированных текстов для метода BagOfWords

Тема 6. Машинное обучение: без учителя. Метрические алгоритмы классификации и кластеризации

Создание кластерного анализа текстов разными методами кластеризации и метриками расстояний: Евклидово расстояние, ближайшие и дальние соседи, расстояние Манхеттона.

Тема 7. Методы сокращения выборки

Разработка Математические модели Lasso, PCA – метод главных компонент

Тема 8. Машинное обучение с учителем.

Виды и типы машинного обучения с учителем, наивный Байесовский классификатор.

Тема 9. Метод опорных векторов (SVM)

Разработка классификатора SVM

Тема 10. Решающие списки и деревья

Создание дерева решений. Интерпретация энтропии Шеннона, индекса Джинни, информационного прироста

Тема 11. Оценка результатов машинного обучения. и выбор моделей.

Разработка функции формальных метрик оценки модели: accuracy, precision, recall, f1-score. Библиотека caret.

Тема 12. Нейронные сети. Типологизация НС. Генерация текстов с помощью НС.

Классификация текстов при помощи нейронной сети transformer LSTM в библиотеке caret. Глубокое обучение, настройка и обучение модели на базе cpu и gpu.

Подготовка к проведению лабораторных работ начинается в начале теоретического изложения изучаемой темы и продолжается по ходу её изучения при освоении материала на занятиях в рамках практических заданий и работе над ним в ходе самостоятельной подготовки дома и в библиотеках. Для качественного выполнения лабораторных работ студентам необходимо:

- 1) повторить теоретический материал по конспекту и учебникам;
- 2) ознакомиться с описанием лабораторной работы;
- 3) в специальной тетради для лабораторных работ записать название и номер работы, перечень необходимого программного обеспечения, подготовить алгоритм или код;
- 4) выяснить цель работы, четко представить себе поставленную задачу и способы её достижения, продумать ожидаемые результаты опытов;
- 5) ответить устно или письменно на контрольные вопросы по изучаемой теме или решить ряд задач;
- 6) изучить порядок выполнения лабораторной работы. Подготовить среду выполнения кода к работе. После проверки правильности алгоритма работы программы преподавателем можно начинать выполнение лабораторной работы.

д) Методические указания по организации самостоятельной работы студентов.

Формы самостоятельной работы студентов разнообразны. Они включают в себя:

- изучение и систематизацию практических и теоретических примеров в рамках выполнения текущих заданий по предмету;
- изучение учебной, научной и методической литературы, материалов периодических изданий с привлечением электронных средств официальной, статистической, периодической и научной информации;
- подготовку докладов и презентаций, написание программного кода и его отладка;
- участие в работе студенческих конференций, комплексных научных исследованиях.

Самостоятельная работа приобщает студентов к научному творчеству, поиску и решению актуальных современных проблем.

Примеры самостоятельной работы студентов:

На основе своего корпуса создайте:

- Частотную матрицу Bag of Words (ключевые слова/части речи/словарь). Веса слов должны быть выражены относительными величинами
- Создайте кластерный анализ по своим данным, используя различные метрики расстояний и методов построения кластеров. Опишите полученный результат, в соответствии со своей гипотезой
- Визуализируйте полученный результат в виде дендрограммы.

Исходный код:

```
library(readtext)
```

```
library(quantda)
```

```
DATA_DIR_s = "D:/tmp/"
```

```
texts_s <- readtext(paste0(DATA_DIR_s, "*.txt"), docvarsfrom = "filenames",  
docvarnames = "document", encoding = "UTF-8")
```

```
library(cluster)
```

```
author<-read.table("author.txt",sep="," ,header = T)
```

```
write.csv(author, "author.csv")
```

```
#correlation
```

```
boxplot(author$all ~ author$Author)
```

```
cor(author[,1:10], method = "pearson")
```

```
cor.test(author$a,author$all, method = "pearson")
```

```
help(cor.test)
```

```
dim(author)
```

```

library(corrplot)
cor_matrix <- cor(author[,1:10],
use = 'complete.obs',
method = "pearson")
corrplot.mixed(cor_matrix, lower = "circle",
upper = "number", tl.pos = "lt",
diag = "u")
#Pearson's Correlation
cor(author[,1:5], author[,1:5], method = 'spearman') # method depends on distribution
cor(author[,1:5], author[,1:5], method = 'pearson')
cor(author[,1:5], author[,1:5], method = 'kendall')
library(GGally)
ggcorr(author[,1:5])
ggcorr(author[,1:5],
nbreaks = 6,
low = "steelblue",
mid = "white",
high = "darkred",
geom = "circle")
ggcorr(author[,1:10],
nbreaks = 6,
label = TRUE,
label_size = 3,
color = "grey50")
library(ggcorrplot)
p.mat <- cor_pmat(author[,1:30] , method = "pearson")
help(cor_pmat)

ggcorrplot(p.mat)
ggcorrplot(p.mat, hc.order = TRUE, type = "lower",
lab = TRUE)
ggcorrplot(p.mat, hc.order = TRUE,
type = "lower", p.mat = p.mat, lab = TRUE) #insig = "blank"

t.test(author$a ~ author$Author)#parametric for 2 classes

kruskal.test(author$also ~ author$Author) #nonparametric for multiple classes

names(author)
dim(author)
table(author$Author)
error<-function(x){((sum(x)-sum(diag(x)))/sum(x))*100}

##CLUSTER

#K-means
fit <- kmeans(author.train2, 3)
# get cluster means
aggregate(author.train2,by=list(fit$cluster),FUN=mean)
# append cluster assignment
mydata <- data.frame(author.train2, fit$cluster)

```

```
# Ward Hierarchical Clustering
d <- dist(mydata, method = "euclidean") # distance matrix
fit <- hclust(d, method="ward.D")
help(hclust)
```

```
plot(fit) # display dendrogram
groups <- cutree(fit, k=2) # cut tree into 5 clusters
# draw dendrogram with red borders around the 5 clusters
rect.hclust(fit, k=3, border="red")
```

```
# K-Means Clustering with 5 clusters
fit <- kmeans(mydata, 2)
```

12. Перечень учебной литературы и ресурсов сети Интернет

а) основная литература:

– Джеймс Г. Введение в статистическое обучение (с примерами на языке R) / Джеймс Г., Уиттон Д., Хасты Т., Тибширани Р. – М.: ДМК, 2016. – 449 с.

– Боярский К. К. Введение в компьютерную лингвистику / Боярский К. К., СПб: ИТМО, 2013. – 73 с.

– Dalgaard P. Introductory Statistics with R New York, NY : : Springer-Verlag New York, , 2008. [Электронный ресурс: <http://dx.doi.org/10.1007/978-0-387-79054-1>]

б) дополнительная литература:

– Thomas Rahr. Data Visualisation with R. Springer International Publishing, New York, 2017.

– Lawrence Leemis. Learning Base R. Lightning Source, 2016.

– Matthias Kohl. Introduction to statistical data analysis with R. bookboon.com, London, 2015.

– Torsten Hothorn and Brian S. Everitt. A Handbook of Statistical Analyses Using R. Chapman & Hall/CRC Press, Boca Raton, Florida, USA, 3rd edition, 2014.

– An Introduction to Statistical Learning: with Applications in R (Springer Texts in Statistics), Corr. 7th printing / G. James, D. Witten, T. Hastie, R. Tibshirani, Springer, 2017,

– Буховец А. Г. Статистический анализ данных в системе R. Учебное пособие / А.Г. Буховец, П.В. Москалев, В.П. Богатова, Т.Я. Бирючинская; Под ред. проф. Буховца А.Г. - Воронеж: ВГАУ, 2010. - 124 с.

– Кабаков Р. R в действии. Анализ и визуализация данных на языке R / Роберт И. Кабаков, – М.: ДМК, 2016. – 587 с.

– Шипунов А. Б. Наглядная статистика. Используем R. / А. Б. Шипунов, Е. М. Балдин, П. А. Волкова, А. И. Коробейников, С. А. Назарова, С. В. Петров, В. Г. Суфиянов, – М.: ДМК, 2017. – 296 с.

– Spector P. Data Manipulation with R / P. Spector New York, NY : Springer Science+Business Media, LLC, 2008

– Ergül Ö. Guide to Programming and Algorithms Using R electronic resource / Ö.Ergül - London : Springer London : Imprint: Springer, 2013. - 182 p.

– Pathak A. Beginning Data Science with R electronic resource / A. Pathak, A. Manas - Springer International Publishing : Imprint: Springer, 2014. - 157 p.

– Larry A. R Recipes electronic resource : A Problem-Solution Approach / Larry A. Pace - Berkeley, CA : Apress : Imprint: Apress, 2014. - 264 p.

– Joshua F. Beginning R electronic resource : An Introduction to Statistical Programming / F. Joshua Wiley, Wiley, Joshua F., Larry A. Pace. - Berkeley, CA : : Apress : Imprint: Apress,, 2015. - 327 p.

– Буховец А. Алгоритмы вычислительной статистики в системе R / А. Буховец - Санкт-Петербург: Лань , 2015. - 147 с.

в) ресурсы сети Интернет:

- открытые онлайн-курсы: <https://online.stanford.edu/lagunita-learning-platform>
- Официальный сайт языка программирования R - www.r-cran.com

13. Перечень информационных технологий

а) лицензионное и свободно распространяемое программное обеспечение:

- Microsoft Office Standart 2013 Russian: пакет программ. Включает приложения: MS Office Word, MS Office Excel, MS Office PowerPoint, MS Office, Windows 7-10;
- публично доступные облачные технологии (Google Docs, Яндекс диск и т.п.).
 - язык программирования R (RStudio) и Python;
 - Программа Mystem.

б) информационные справочные системы:

- Электронный каталог Научной библиотеки ТГУ – <http://chamo.lib.tsu.ru/search/query?locale=ru&theme=system>
- Электронная библиотека (репозиторий) ТГУ – <http://vital.lib.tsu.ru/vital/access/manager/Index>

в) профессиональные базы данных:

- Университетская информационная система РОССИЯ – <https://uisrussia.msu.ru/>
- Единая межведомственная информационно-статистическая система (ЕМИСС) – <https://www.fedstat.ru/>
- Справка ПО и библиотек R-CRAN <https://cran.r-project.org/>

14. Материально-техническое обеспечение

Аудитории для проведения занятий лекционного типа.

Аудитории для проведения занятий семинарского типа, индивидуальных и групповых консультаций, текущего контроля и промежуточной аттестации.

Помещения для самостоятельной работы, оснащенные компьютерной техникой и доступом к сети Интернет, в электронную информационно-образовательную среду и к информационным справочным системам.

Лаборатории, оборудованные компьютерами (не ниже i3, RAM 8Gb), проектором

Аудитории для проведения занятий лекционного и семинарского типа индивидуальных и групповых консультаций, текущего контроля и промежуточной аттестации в смешенном формате («Актру»).

15. Информация о разработчиках

Степаненко Андрей Александрович, НИ Томский государственный университет, ассистент кафедры общей, компьютерной и когнитивной лингвистики ТГУ.