

# lab4

Jiajun Zhang

April 14, 2019

```
library(RSQLite)
library(DBI)
# setwd("E:/2019Spring/STAT240/lab4")
dbcon = dbConnect(SQLite(), dbname="stat240Week4lab2019.sqlite")
dbListTables(dbcon)

## [1] "CA"          "Olymp_meds" "POP2006"     "POP2011"     "Pokem"
## [6] "Vanpoke"     "Winter0"     "tickets"     "zip"

dbGetQuery(dbcon, "PRAGMA table_info('POP2006')")

##   cid                                     name
## 1   0                                     Geographic_name
## 2   1 Incompletely_enumerated_Indian_reserves_and_Indian_settlements__2006
## 3   2                                     Population__2006
## 4   3                                     Total_private_dwellings__2006
## 5   4 Private_dwellings_occupied_by_usual_residents__2006
##      type notnull dflt_value pk
## 1   TEXT      0         NA  0
## 2 INTEGER      0         NA  0
## 3 INTEGER      0         NA  0
## 4 INTEGER      0         NA  0
## 5 INTEGER      0         NA  0
```

## Question1

(a)

```
# dbReadTable(dbcon, "Vanpoke")
dbGetQuery(dbcon, "SELECT DISTINCT name FROM Vanpoke ORDER BY name DESC")

##      name
## 1 Wigglytuff
## 2 Whiscash
## 3 Wailmer
## 4 Vileplume
## 5 Vigoroth
## 6 Victreebel
## 7 Venusaur
## 8 Vaporeon
## 9 Ursaring
## 10 Treecko
## 11 Trapinch
## 12 Torchic
## 13 Togetic
## 14 Teddiursa
## 15 Swellow
## 16 Solrock
```

## 17 Snorlax  
## 18 Slakoth  
## 19 Sharpedo  
## 20 Sealeo  
## 21 Seadra  
## 22 Scyther  
## 23 Rhyhorn  
## 24 Rhydon  
## 25 Ralts  
## 26 Pupitar  
## 27 Primeape  
## 28 Porygon  
## 29 Poliwrath  
## 30 Poliwhirl  
## 31 Piloswine  
## 32 Pelipper  
## 33 Onix  
## 34 Omastar  
## 35 Ninetales  
## 36 Nidoqueen  
## 37 Nidoking  
## 38 Muk  
## 39 Mudkip  
## 40 Mightyena  
## 41 Metang  
## 42 Masquerain  
## 43 Marshtomp  
## 44 Mareep  
## 45 Magikarp  
## 46 Machop  
## 47 Machoke  
## 48 Machop  
## 49 Luvdisc  
## 50 Loudred  
## 51 Lotad  
## 52 Lombre  
## 53 Lickitung  
## 54 Larvitar  
## 55 Lapras  
## 56 Lairon  
## 57 Kirlia  
## 58 Kadabra  
## 59 Jolteon  
## 60 Houndoom  
## 61 Hitmontop  
## 62 Hitmonlee  
## 63 Hitmonchan  
## 64 Haunter  
## 65 Hariyama  
## 66 Grovyle  
## 67 Grimer  
## 68 Graveler  
## 69 Golem  
## 70 Geodude

```
## 71      Gengar
## 72      Flaaffy
## 73      Feebas
## 74      Exeggutor
## 75      Dusclops
## 76      Dratini
## 77      Dragonite
## 78      Dragonair
## 79      Donphan
## 80      Dewgong
## 81      Crawdaunt
## 82      Combusken
## 83      Cloyster
## 84      Clefable
## 85      Claydol
## 86      Chimecho
## 87      Chansey
## 88      Camerupt
## 89      Cacturne
## 90      Breloom
## 91      Beldum
## 92      Banette
## 93      Bagon
## 94      Aron
## 95      Armaldo
## 96      Alakazam
## 97      Aerodactyl
## 98      Abra
```

(b)

```
# dbReadTable(dbcon, "Olymp_meds")
dbGetQuery(dbcon, "SELECT DISTINCT NOC, Athlete FROM Olymp_meds WHERE Athlete
LIKE 'Zh%' ORDER BY NOC")
```

```
##      NOC      Athlete
## 1  BLR      ZHUKOVA, Inna
## 2  CHN      ZHOU, Ji-Hong
## 3  CHN      ZHU, Jian-Hua
## 4  CHN      ZHANG, Hui
## 5  CHN      ZHANG, Yue-Qin
## 6  CHN      ZHENG, Haixia
## 7  CHN      ZHOU, Ping
## 8  CHN      ZHOU, Qiu-Rui
## 9  CHN      ZHANG, Pei-Jun
## 10 CHN      ZHANG, Wei-Hong
## 11 CHN      ZHU, Jue-Feng
## 12 CHN      ZHANG, Rong-Fang
## 13 CHN      ZHENG, Mei-Zhu
## 14 CHN      ZHOU, Xiao-Lan
## 15 CHN      ZHU, Ling
## 16 CHN      ZHOU, Pei-Shun
## 17 CHN      ZHUANG, Yong
```

## 18 CHN	ZHANG, Xiang-Hua
## 19 CHN	ZHANG, Ya-Li
## 20 CHN	ZHOU, Shouying
## 21 CHN	ZHOU, Xiu-Hua
## 22 CHN	ZHAO, Hong
## 23 CHN	ZHAO, Kun
## 24 CHN	ZHAN, Shuping
## 25 CHN	ZHENG, Dongmei
## 26 CHN	ZHENG, Xiulin
## 27 CHN	ZHUANG, Xiaoyan
## 28 CHN	ZHANG, Di
## 29 CHN	ZHANG, Xiaodong
## 30 CHN	ZHANG, Shan
## 31 CHN	ZHAO, Lihong
## 32 CHN	ZHONG, Honglian
## 33 CHN	ZHANG, Jinjing
## 34 CHN	ZHANG, Xiuyun
## 35 CHN	ZHANG, Bing
## 36 CHN	ZHANG, Chunfang
## 37 CHN	ZHU, Yunying
## 38 CHN	ZHANG, Xiangsen
## 39 CHN	ZHAN, Xugang
## 40 CHN	ZHANG, Jun
## 41 CHN	ZHENG, Lihui
## 42 CHN	ZHANG, Xiangxiang
## 43 CHN	ZHU, Yingwen
## 44 CHN	ZHANG, Juan Juan
## 45 CHN	ZHANG, Jiewen
## 46 CHN	ZHOU, Mi
## 47 CHN	ZHANG, Ning
## 48 CHN	ZHANG, Nan
## 49 CHN	ZHU, Qinan
## 50 CHN	ZHANG, Yining
## 51 CHN	ZHANG, Na
## 52 CHN	ZHANG, Ping
## 53 CHN	ZHANG, Yuehong
## 54 CHN	ZHAO, Ruirui
## 55 CHN	ZHOU, Suhong
## 56 CHN	ZHANG, Guozheng
## 57 CHN	ZHOU, Luxin
## 58 CHN	ZHANG, Lin
## 59 CHN	ZHAO, Jing
## 60 CHN	ZHOU, Yafei
## 61 CHN	ZHU, Qianwei
## 62 CHN	ZHANG, Xiaohuan
## 63 CHN	ZHANG, Wenxiu
## 64 CHN	ZHOU, Chunxiu
## 65 CHN	ZHANG, Yawen
## 66 CHN	ZHANG, Zhilei
## 67 CHN	ZHANG, Xiaoping
## 68 CHN	ZHONG, Man
## 69 CHN	ZHANG, Shuo
## 70 CHN	ZHANG, Yimeng
## 71 CHN	ZHAO, Yudiao

## 72	CHN	ZHOU, Wanfeng
## 73	CHN	ZHANG, Yangyang
## 74	CHN	ZHU, Guo
## 75	CHN	ZHENG, Jie
## 76	CHN	ZHANG, Xi
## 77	ESP	ZHIVANEVSKAYA, Nina Alexandrovna
## 78	EUN	ZHIVANEVSKAYA, Nina Alexandrovna
## 79	EUN	ZHIRKO, Elena
## 80	KAZ	ZHITKEYEV, Askhat
## 81	UKR	ZHUPINA, Olena
## 82	UKR	ZHOVNIR, Olha
## 83	URS	ZHILIN, Georgi
## 84	URS	ZHDANOVICH, Viktor
## 85	URS	ZHABOTINSKY, Leonid
## 86	URS	ZHARKOVA-MASLAKOVA, Liudmila
## 87	URS	ZHARMUKHAMEDOV, Alzhan
## 88	URS	ZHELEZNYAK, Yakov
## 89	URS	ZHIGILY, Vladimir
## 90	URS	ZHULPA, Robertas
## 91	URS	ZHELANOV, Sergei
## 92	URS	ZHUK, Aleksei
## 93	URS	ZHULINA, Valentina
## 94	URS	ZHUPEEVA, Elena
## 95	URS	ZHIROVA, Marina
## 96	URS	ZHEVKOVA, Olga

## Question2

(a)

```
poke_out=dbGetQuery(dbcon, "SELECT name, count(name) AS PokemonCounts FROM Vanpoke GROUP BY name")
poke_out
```

##	name	PokemonCounts
## 1	Abra	514
## 2	Aerodactyl	6
## 3	Alakazam	1
## 4	Armaldo	7
## 5	Aron	668
## 6	Bagon	73
## 7	Banette	27
## 8	Beldum	119
## 9	Breloom	22
## 10	Cacturne	10
## 11	Camerupt	2
## 12	Chansey	99
## 13	Chimecho	6
## 14	Claydol	140
## 15	Clefable	17
## 16	Cloyster	10
## 17	Combusken	1
## 18	Crawdaunt	2
## 19	Dewgong	13
## 20	Donphan	25
## 21	Dragonair	1
## 22	Dragonite	1
## 23	Dratini	10
## 24	Dusclops	1
## 25	Exeggutor	1
## 26	Feebas	86
## 27	Flaaffy	1
## 28	Gengar	4
## 29	Geodude	705
## 30	Golem	3
## 31	Graveler	39
## 32	Grimer	29
## 33	Grovyle	2
## 34	Hariyama	78
## 35	Haunter	55
## 36	Hitmonchan	5
## 37	Hitmonlee	7
## 38	Hitmontop	4
## 39	Houndoom	1
## 40	Jolteon	2
## 41	Kadabra	11
## 42	Kirlia	3
## 43	Lairon	82
## 44	Lapras	7
## 45	Larvitar	21
## 46	Lickitung	5

## 47	Lombre	1
## 48	Lotad	424
## 49	Loudred	333
## 50	Luvdisc	2586
## 51	Machamp	2
## 52	Machoke	29
## 53	Machop	2086
## 54	Magikarp	589
## 55	Mareep	6
## 56	Marshtomp	9
## 57	Masquerain	33
## 58	Metang	2
## 59	Mightyena	146
## 60	Mudkip	1091
## 61	Muk	2
## 62	Nidoking	9
## 63	Nidoqueen	5
## 64	Ninetales	13
## 65	Omastar	1
## 66	Onix	52
## 67	Pelipper	4
## 68	Piloswine	105
## 69	Poliwhirl	1
## 70	Poliwrath	2
## 71	Porygon	4
## 72	Primeape	5
## 73	Pupitar	1
## 74	Ralts	321
## 75	Rhydon	2
## 76	Rhyhorn	297
## 77	Scyther	18
## 78	Seadra	8
## 79	Sealeo	64
## 80	Sharpedo	5
## 81	Slakoth	207
## 82	Snorlax	2
## 83	Solrock	149
## 84	Swellow	246
## 85	Teddiursa	306
## 86	Togetic	20
## 87	Torchic	122
## 88	Trapinch	32
## 89	Treecko	553
## 90	Ursaring	8
## 91	Vaporeon	3
## 92	Venusaur	1
## 93	Victreebel	5
## 94	Vigoroth	3
## 95	Vileplume	13
## 96	Wailmer	1296
## 97	Whiscash	51
## 98	Wigglytuff	23

(b)

```
poke_out[poke_out[, "PokemonCounts"]==5,]
```

```
##           name PokemonCounts
## 36 Hitmonchan             5
## 46 Lickitung              5
## 63 Nidoqueen              5
## 72 Primeape               5
## 80 Sharpedo               5
## 93 Victreebel             5
```

(c)

```
sql_moveset = "SELECT DISTINCT moveset, name FROM Vanpoke WHERE name == 'Hitmonchan'
OR name == 'Lickitung' OR name == 'Nidoqueen' OR name == 'Primeape' OR name == 'Sharpedo'
OR name == 'Victreebel' ORDER BY moveset"
moveset_out=dbSendQuery(dbcon, sql_moveset)
dbFetch(moveset_out, 2)
```

```
##           moveset      name
## 1 Acid - Leaf Blade Victreebel
## 2 Acid - Sludge Bomb Victreebel
```

```
dbFetch(moveset_out, 2)
```

```
##           moveset      name
## 1 Bite - Earthquake Nidoqueen
## 2 Bite - Sludge Wave Nidoqueen
```

```
dbFetch(moveset_out, 2)
```

```
##           moveset      name
## 1 Bite - Stone Edge Nidoqueen
## 2 Bullet Punch - Ice Punch Hitmonchan
```

```
dbFetch(moveset_out, -1)
```

```
##           moveset      name
## 1 Bullet Punch - Thunder Punch Hitmonchan
## 2 Counter - Ice Punch Hitmonchan
## 3 Counter - Low Sweep Primeape
## 4 Counter - Night Slash Primeape
## 5 Lick - Power Whip Lickitung
## 6 Lick - Stomp Lickitung
## 7 Low Kick - Close Combat Primeape
## 8 Low Kick - Low Sweep Primeape
## 9 Low Kick - Night Slash Primeape
## 10 Poison Jab - Earthquake Nidoqueen
## 11 Razor Leaf - Leaf Blade Victreebel
## 12 Zen Headbutt - Hyper Beam Lickitung
## 13 Zen Headbutt - Power Whip Lickitung
```

```
dbClearResult(moveset_out)
```



### Question3

(a)

```
dbGetQuery(dbcon, "SELECT * FROM Winter0 WHERE Athlete=='Tessa VIRTUE'")
```

```
##           place year Country Medal           Event Athlete Record
## 1   Vancouver 2010   Canada   Gold ice dancing mixed Tessa VIRTUE
## 2     Sochi 2014   Canada Silver ice dancing mixed Tessa VIRTUE
## 3     Sochi 2014   Canada Silver      pairs mixed Tessa VIRTUE
## 4 PyeongChang 2018   Canada   Gold ice1 dancing mixed Tessa VIRTUE
## 5 PyeongChang 2018   Canada   Gold      pairs1 mixed Tessa VIRTUE
```

```
sql_ins2 = "INSERT INTO Winter0 (place,year,Country,Medal,Event,Athlete,Record) VALUES
('Sochi', '2014', 'Canada', 'Silver','ice dancing mixed','Tessa VIRTUE',''),
('Sochi', '2014', 'Canada', 'Silver','ice dancing mixed','Scott MOIR',''),
('Sochi', '2014', 'Canada', 'Silver','pairs mixed','Tessa VIRTUE',''),
('Sochi', '2014', 'Canada', 'Silver','pairs mixed','Scott MOIR',''),
('PyeongChang', '2018', 'Canada', 'Gold','ice1 dancing mixed','Tessa VIRTUE',''),
('PyeongChang', '2018', 'Canada', 'Gold','ice1 dancing mixed','Scott MOIR',''),
('PyeongChang', '2018', 'Canada', 'Gold','pairs1 mixed','Tessa VIRTUE',''),
('PyeongChang', '2018', 'Canada', 'Gold','pairs1 mixed','Scott MOIR','')"
```

```
dbSendQuery(dbcon, sql_ins2)
```

```
## <SQLiteResult>
```

```
## SQL INSERT INTO Winter0 (place,year,Country,Medal,Event,Athlete,Record) VALUES
## ('Sochi', '2014', 'Canada', 'Silver','ice dancing mixed','Tessa VIRTUE',''),
## ('Sochi', '2014', 'Canada', 'Silver','ice dancing mixed','Scott MOIR',''),
## ('Sochi', '2014', 'Canada', 'Silver','pairs mixed','Tessa VIRTUE',''),
## ('Sochi', '2014', 'Canada', 'Silver','pairs mixed','Scott MOIR',''),
## ('PyeongChang', '2018', 'Canada', 'Gold','ice1 dancing mixed','Tessa VIRTUE',''),
## ('PyeongChang', '2018', 'Canada', 'Gold','ice1 dancing mixed','Scott MOIR',''),
## ('PyeongChang', '2018', 'Canada', 'Gold','pairs1 mixed','Tessa VIRTUE',''),
## ('PyeongChang', '2018', 'Canada', 'Gold','pairs1 mixed','Scott MOIR','')
## ROWS Fetched: 0 [complete]
##           Changed: 8
```

```
dbGetQuery(dbcon, "SELECT * FROM Winter0 WHERE Athlete == 'Tessa VIRTUE' OR
Athlete == 'Scott MOIR' ")
```

```
##           place year Country Medal           Event Athlete Record
## 1   Vancouver 2010   Canada   Gold ice dancing mixed Tessa VIRTUE
## 2   Vancouver 2010   Canada   Gold ice dancing mixed  Scott MOIR
## 3     Sochi 2014   Canada Silver ice dancing mixed Tessa VIRTUE
## 4     Sochi 2014   Canada Silver ice dancing mixed  Scott MOIR
## 5     Sochi 2014   Canada Silver      pairs mixed Tessa VIRTUE
## 6     Sochi 2014   Canada Silver      pairs mixed  Scott MOIR
## 7 PyeongChang 2018   Canada   Gold ice1 dancing mixed Tessa VIRTUE
## 8 PyeongChang 2018   Canada   Gold ice1 dancing mixed  Scott MOIR
## 9 PyeongChang 2018   Canada   Gold      pairs1 mixed Tessa VIRTUE
## 10 PyeongChang 2018   Canada   Gold      pairs1 mixed  Scott MOIR
## 11     Sochi 2014   Canada Silver ice dancing mixed Tessa VIRTUE
## 12     Sochi 2014   Canada Silver ice dancing mixed  Scott MOIR
## 13     Sochi 2014   Canada Silver      pairs mixed Tessa VIRTUE
## 14     Sochi 2014   Canada Silver      pairs mixed  Scott MOIR
```

```
## 15 PyeongChang 2018 Canada Gold ice1 dancing mixed Tessa VIRTUE
## 16 PyeongChang 2018 Canada Gold ice1 dancing mixed Scott MOIR
## 17 PyeongChang 2018 Canada Gold pairs1 mixed Tessa VIRTUE
## 18 PyeongChang 2018 Canada Gold pairs1 mixed Scott MOIR
```

(b)

```
sql_ins3 = "INSERT INTO Winter0 (place, year, Country, Medal, Event, Athlete)
VALUES ('Calgary', '1988', 'China', 'Gold', '500m men', 'Jiajun Zhang'),
('Vancouver', '2010', 'China', 'Gold', '1000m men', 'Jiajun Zhang'),
('Sochi', '2014', 'China', 'Gold', '1500m men', 'Jiajun Zhang')"
dbSendQuery(dbcon, sql_ins3)
```

```
## <SQLiteResult>
## SQL INSERT INTO Winter0 (place, year, Country, Medal, Event, Athlete)
## VALUES ('Calgary', '1988', 'China', 'Gold', '500m men', 'Jiajun Zhang'),
## ('Vancouver', '2010', 'China', 'Gold', '1000m men', 'Jiajun Zhang'),
## ('Sochi', '2014', 'China', 'Gold', '1500m men', 'Jiajun Zhang')
## ROWS Fetched: 0 [complete]
## Changed: 3
```

```
dbGetQuery(dbcon, "SELECT * FROM Winter0 WHERE Athlete == 'Jiajun Zhang'")
```

```
## Warning: Closing open result set, pending rows
```

```
##      place year Country Medal      Event      Athlete Record
## 1   Calgary 1988  China  Gold   500m men Jiajun Zhang  <NA>
## 2     Sochi 2014  China  Gold  1500m men Jiajun Zhang  <NA>
## 3   Calgary 1988  China  Gold   500m men Jiajun Zhang  <NA>
## 4 Vancouver 2010  China  Gold  1000m men Jiajun Zhang  <NA>
## 5     Sochi 2014  China  Gold  1500m men Jiajun Zhang  <NA>
## 6   Calgary 1988  China  Gold   500m men Jiajun Zhang  <NA>
## 7 Vancouver 2010  China  Gold  1000m men Jiajun Zhang  <NA>
## 8     Sochi 2014  China  Gold  1500m men Jiajun Zhang  <NA>
```

```
# sql_del = "DELETE FROM Winter0 WHERE Athlete == 'Jiajun Zhang'"
# dbSendQuery(dbcon, sql_del)
```

(c)

```
dbGetQuery(dbcon, "SELECT * FROM Winter0 WHERE Athlete LIKE '%Zhang'")
```

```
##      place year Country Medal      Event
## 1 Lillehammer 1994 People's Republic of China Silver      500m women
## 2      Turin 2006 People's Republic of China Silver      pairs mixed
## 3      Turin 2006 People's Republic of China Silver      pairs mixed
## 4   Vancouver 2010 People's Republic of China   Gold 3000m relay women
## 5     Calgary 1988           China   Gold      500m men
## 6     Sochi 2014           China   Gold     1500m men
## 7     Calgary 1988           China   Gold      500m men
## 8   Vancouver 2010           China   Gold     1000m men
## 9     Sochi 2014           China   Gold     1500m men
## 10    Calgary 1988           China   Gold      500m men
## 11   Vancouver 2010           China   Gold     1000m men
```

## 12	Sochi 2014	China	Gold	1500m men
##	Athlete Record			
## 1	Yanmei ZHANG			
## 2	Hao ZHANG			
## 3	Dan ZHANG			
## 4	Hui ZHANG			
## 5	Jiajun Zhang	<NA>		
## 6	Jiajun Zhang	<NA>		
## 7	Jiajun Zhang	<NA>		
## 8	Jiajun Zhang	<NA>		
## 9	Jiajun Zhang	<NA>		
## 10	Jiajun Zhang	<NA>		
## 11	Jiajun Zhang	<NA>		
## 12	Jiajun Zhang	<NA>		

## Question4

(a)

```
sql_del="DELETE FROM WinterO WHERE year=='2010' AND Athlete == 'Jiajun Zhang'"
dbSendQuery(dbcon, sql_del)
```

```
## <SQLiteResult>
##   SQL  DELETE FROM WinterO WHERE year=='2010' AND Athlete == 'Jiajun Zhang'
##   ROWS Fetched: 0 [complete]
##           Changed: 2
```

(b)

```
dbGetQuery(dbcon, "SELECT * FROM WinterO WHERE Athlete == 'Jiajun Zhang'")
```

```
## Warning: Closing open result set, pending rows
```

```
##   place year Country Medal   Event   Athlete Record
## 1 Calgary 1988   China  Gold  500m men Jiajun Zhang  <NA>
## 2  Sochi  2014   China  Gold 1500m men Jiajun Zhang  <NA>
## 3 Calgary 1988   China  Gold  500m men Jiajun Zhang  <NA>
## 4  Sochi  2014   China  Gold 1500m men Jiajun Zhang  <NA>
## 5 Calgary 1988   China  Gold  500m men Jiajun Zhang  <NA>
## 6  Sochi  2014   China  Gold 1500m men Jiajun Zhang  <NA>
```

## Question5

(a)

```
sql_poke1="SELECT Vanpoke.cp, Vanpoke.level, Pokem.Type_1, Pokem.Type_2
FROM Pokem INNER JOIN Vanpoke ON Pokem.Name=Vanpoke.name"
df=dbGetQuery(dbcon, sql_poke1)
dim(df)
```

```
## [1] 14222      4
```

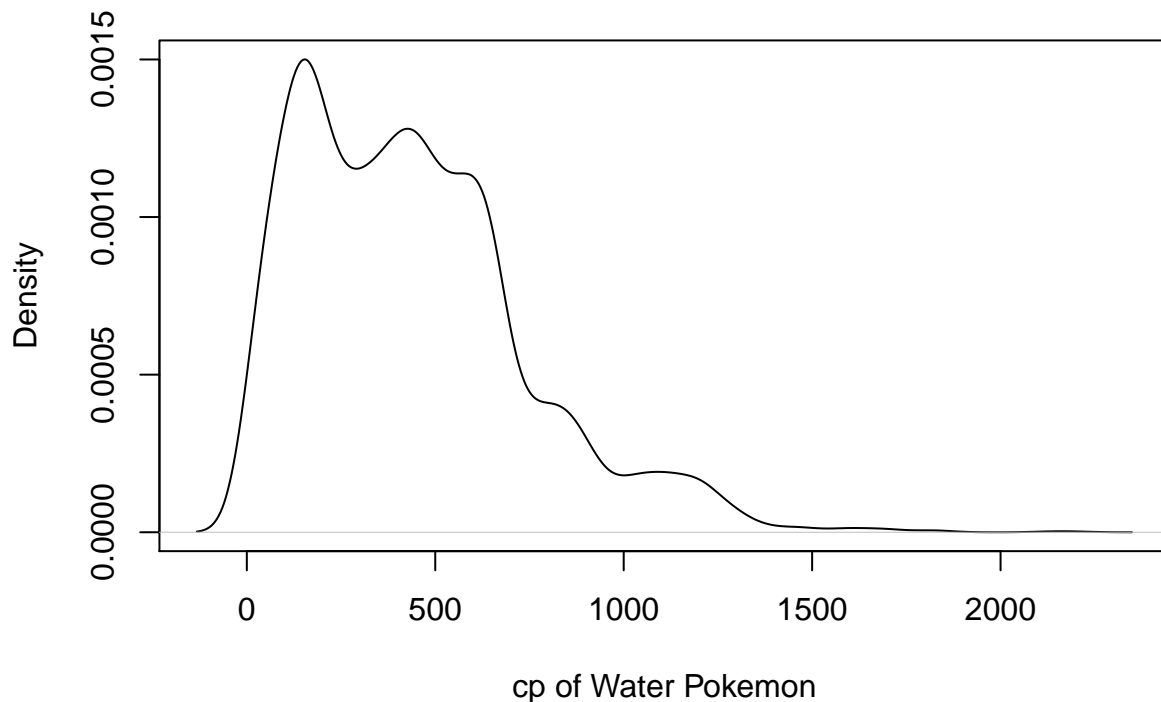
```
head(df)
```

```
##      cp level Type_1 Type_2
## 1 1090    15  Grass Poison
## 2  659    10 Poison Ground
## 3  721    11 Poison Ground
## 4  795    12 Poison Ground
## 5 1063    16 Poison Ground
## 6 2093    33 Poison Ground
```

(b)

```
sql_vanpoke="SELECT Vanpoke.cp, Pokem.Type_1, Pokem.Type_2 FROM Pokem
INNER JOIN Vanpoke ON Pokem.Name=Vanpoke.name WHERE
Pokem.Type_1=='Water' OR Pokem.Type_2=='Water'"
df1=dbGetQuery(dbcon, sql_vanpoke)
plot(density(df1$cp), xlab="cp of Water Pokemon",
      main="Density Plot of cp Where Pokemon is Water Type")
```

### Density Plot of cp Where Pokemon is Water Type



(c)

```
sql_poke2="SELECT Poken.Type_1, Poken.Type_2, Vanpoke.cp
FROM Poken INNER JOIN Vanpoke ON Poken.Name=Vanpoke.name"
df2=dbGetQuery(dbcon, sql_poke2)
plot(density(df2$cp), xlab="cp of All Pokemon", ylim = c(0, 3e-03),
     main="Density Plot of cp by Pokemon Types")
```

```
unique_type=unique(df2$Type_1)
unique(df2$Type_1)
```

```
## [1] "Grass"    "Poison"   "Fairy"    "Fire"     "Normal"   "Fighting"
## [7] "Water"    "Psychic"  "Rock"     "Ghost"    "Ground"   "Bug"
## [13] "Electric" "Dragon"   "Ice"      "Dark"     "Steel"
```

```
#Type1 has 17 total different types
unique(df2$Type_2)
```

```
## [1] "Poison"   "Ground"   ""          "Fairy"    "Fighting" "Ice"
## [7] "Psychic"  "Rock"     "Flying"    "Water"    "Fire"     "Grass"
## [13] "Dark"     "Bug"
```

```
#Flying from type2 is not in type1
```

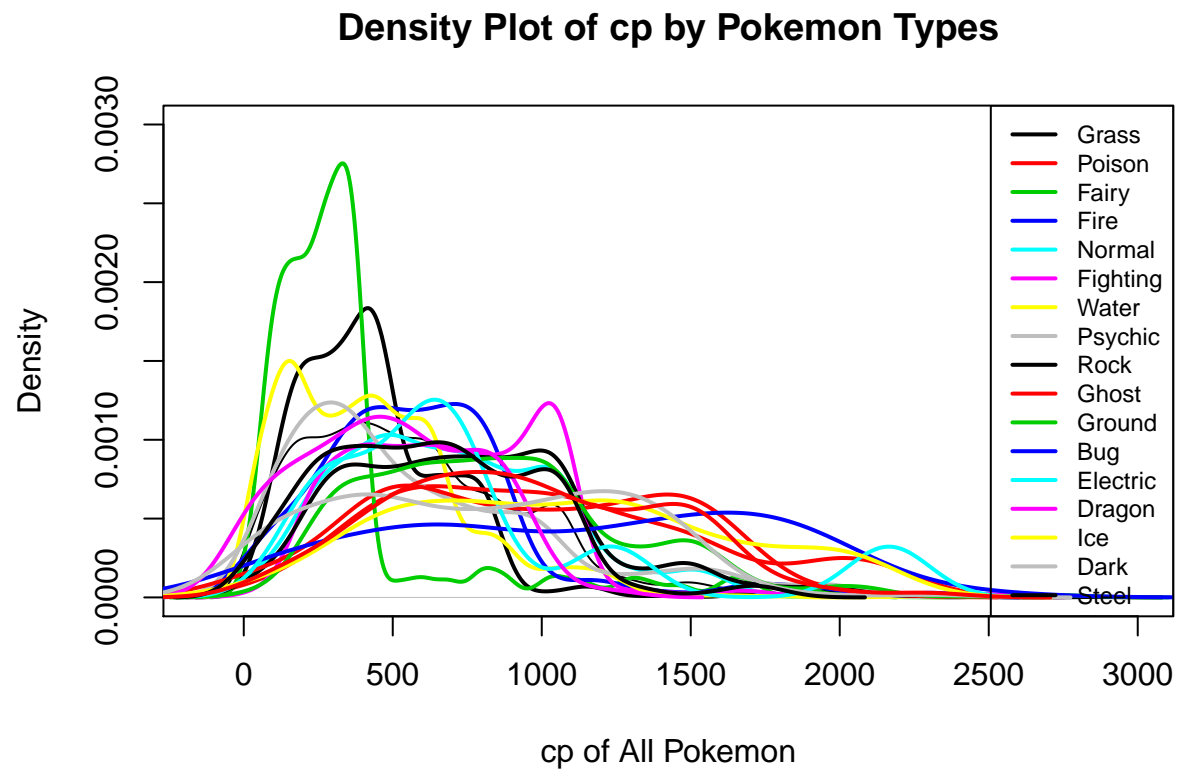
```
for(poke_type in seq_along(unique_type)){
  lines(density(df2$cp[df2$Type_1==unique_type[poke_type] | df2$Type_2==unique_type[poke_type]]), col=)
```

```

}
lines(density(df2$cp[df2$Type_2=="Flying"]), col=18, lwd=2)

legend("topright", col=1:18, lwd=2, legend=c(unique_type, "Flying"), cex=0.75)

```



```
dbDisconnect(dbcon)
```