R Lab3: Data Management(II): Data Cleanning and Visualization

QSS,2025

Zhaopeng Qu

HNC

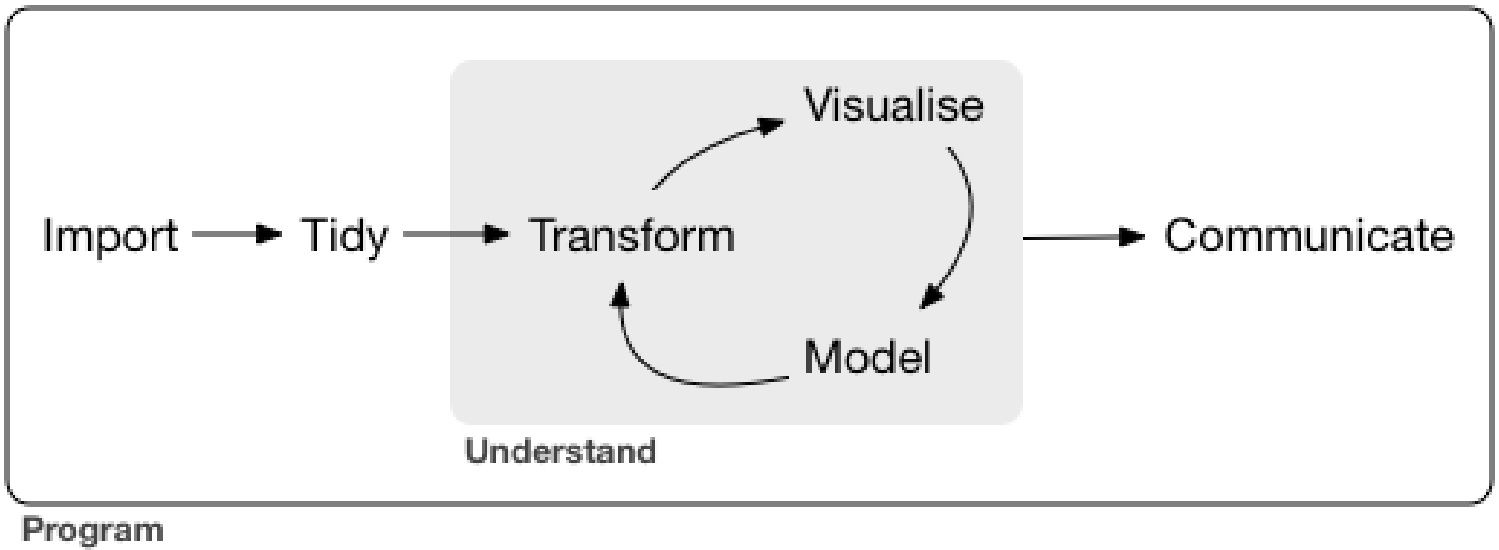
March 17 2025

Section 1

Review Last Lecture

Review: Tidy Data and the Workflow

- Variable and Data Types in R
- Data Structure in R
- Two Models to clean data in R
  - Basic R
  - `Tidyverse`
- The **workflow** of data analysis is as follows



Review: `tidy` data

- Import Data
- **Transform and Relate Data(Structure)**
- Wrangling Data
  - generate new variables
  - select variables and observations
  - relating data(join or merge datasets)
  - Transform data
- Visualizing Data(Figures and Tables)
- **Communicating Data(Rmarkdown)**

Today: `tidy` data and Visualization

- Improve data quality(observations):
  - Missing values
  - Duplicates
  - Outliers
  - Labels
- Visualizing Data
  - Figures
  - Tables

Section 2

Introduction to Missing values

Introduction

- Working with *real-world* data = *working with missing data*.
- Missing values are values that *should have been recorded but were not*.
- Missing values in R
  - Explicitly: **NA** = *Not Available* or other forms
  - Implicitly: simply not present in the data

Missing values in Data

- Find and unify missing values
- Summarize missing values
- Deal with missing values

Find missing values

- Whether easy to tell
  - explicit
  - implicit(panel)
- Whether united forms

A Special Data Case

```
df <- read_csv("Data/telecom.csv")
glimpse(df)
```

```
## Rows: 10
## Columns: 5
## $ customerID     <chr> "7590-VHVEG", "5575-GNVDE", "3668-QPYBK", "7795-C
## $ MonthlyCharges <dbl> 29.85, 56.95, NA, 42.30, 70.70, NA, 89.10, NA, 10
## $ TotalCharges   <chr> "109.9", "na", "108.15", "1840.75", NA, "820.5",
## $ PaymentMethod  <chr> "Electronic check", "Mailed check", "--", "Bank t
## $ Churn          <chr> "yes", "yes", "yes", "no", "no", "yes", "no", "ye
```

Find missing value commands

- `is.na()`
- `any_na()`
- `n_miss_row()`

Missing values in Data

```
df$MonthlyCharges
```

```
##   [1]   29.85   56.95      NA   42.30   70.70     NaN   89.10      N
```

```
is.na(df$MonthlyCharges)
```

```
##   [1] FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE
```

- R only identified "NA" as a missing value.

```
is.nan(df$MonthlyCharges)
```

```
##   [1] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
```

- Missing values in R
  - **NaN** = *Not a Number*

Missing Values in Data: More

- “Missing”
- “-99999”
- “N/A”
- “-”
- “.”
- ” ”
- ...
- “.d” : did not know
- “.e” : inappropriate
- “.r” : refused

CASE: Missing Values

```
head(df)
```

```
## # A tibble: 6 x 5
##   customerID MonthlyCharges TotalCharges PaymentMethod    C
##   <chr>                <dbl> <chr>        <chr>            <
## 1 7590-VHVEG            29.8 109.9        Electronic check y
## 2 5575-GNVDE            57.0 na           Mailed check     y
## 3 3668-QPYBK             NA  108.15       --               y
## 4 7795-CFOCW            42.3 1840.75      Bank transfer    n
## 5 9237-HQITU            70.7 <NA>         Electronic check n
## 6 9305-CDSKC           NaN   820.5        --               y
```

Unified forms of missing values

```
#install.packages("naniar")
```

- `naniar:: replace_with_na`: turn missing values into NA
  - `replace_with_na`
  - `replace_with_na_all()`
  - `replace_with_na_at()`
  - `replace_with_na_if()`
- `tidyr::replace_na`: turn missing values(NA) into a value

Replacing missing values with “NA”

```
df$PaymentMethod
```

```
##  [1] "Electronic check" "Mailed check"      "--"
##  [5] "Electronic check" "--"                "Credit card"
##  [9] "Electronic check" "Electronic check"
```

Missing value in R: Replacing missing values

```
df %>%
replace_with_na(replace = list(PaymentMethod=c("","--")))
```

```
## # A tibble: 10 x 5
##    customerID MonthlyCharges TotalCharges PaymentMethod
##    <chr>               <dbl> <chr>        <chr>
##  1 7590-VHVEG           29.8 109.9        Electronic check
##  2 5575-GNVDE           57.0 na           Mailed check
##  3 3668-QPYBK            NA  108.15       <NA>
##  4 7795-CFOCW           42.3 1840.75      Bank transfer
##  5 9237-HQITU           70.7 <NA>         Electronic check
##  6 9305-CDSKC            NaN 820.5        <NA>
##  7 1452-KIOVK           89.1 1949.4       Credit card
##  8 6713-OKOMC            NA  N/A          <NA>
##  9 7892-POOKP           105. 3046.05      Electronic check
## 10 8451-AJOMK           54.1 354.95       Electronic check
```

Replacing missing values

- Practice by your self
  - replace all potential missing values with “NA”

Analysis Missing Values

- Before dealing with missing values in the data, it's important to find them and figure out why they exist in the first place
  - **MCAR**: Missing Completely at Random
  - **MAR**: Missing At Random
  - **MNAR**: Missing Not At Random

Analysis Missing Values: MCAR

Missingness has no association with any data you have observed, or not observed.

test	vacation
NA	TRUE
11.533340	FALSE
10.126115	TRUE
NA	FALSE
NA	TRUE
8.551881	FALSE
NA	FALSE
NA	TRUE
10.608264	TRUE
8.611877	TRUE

Analysis Missing Values: MAR

Missingness depends on data observed, but not data observed

Implications:

- Impute
- Deleting observations not ideal, may lead to bias

test	vacation	depression
NA	TRUE	87.93109
11.533340	FALSE	40.02708
10.126115	TRUE	48.62883
NA	FALSE	88.21743
NA	TRUE	90.29282
8.551881	FALSE	44.77343
NA	FALSE	89.48865
NA	TRUE	89.99209
10.608264	TRUE	45.56832
8.611877	TRUE	42.41686

Analysis Missing Values: MNAR

Missingness of the response is related to an unobserved value relevant to the assessment of interest.

Implications:

- Data will be biased from deletion and imputation
- Inference can be limited, proceed with caution.

test	vacation	depression
NA	TRUE	NA
11.533340	FALSE	11.533340
10.126115	TRUE	10.126115
NA	FALSE	NA
NA	TRUE	NA
8.551881	FALSE	8.551881
NA	FALSE	NA
NA	TRUE	NA
10.608264	TRUE	10.608264
8.611877	TRUE	8.611877

Summary of missing values

- R Function
  - `n_miss`
  - `n_complete`
  - `miss_var_summary`
  - `miss_case_summary`

Summary of missing values in R

```
airquality
```

```
##     Ozone Solar.R Wind Temp Month Day
## 1      41     190  7.4   67     5   1
## 2      36     118  8.0   72     5   2
## 3      12     149 12.6   74     5   3
## 4      18     313 11.5   62     5   4
## 5      NA      NA 14.3   56     5   5
## 6      28      NA 14.9   66     5   6
## 7      23     299  8.6   65     5   7
## 8      19      99 13.8   59     5   8
## 9       8      19 20.1   61     5   9
## 10     NA     194  8.6   69     5  10
## 11      7      NA  6.9   74     5  11
## 12     16     256  9.7   69     5  12
## 13     11     290  9.2   66     5  13
## 14     14     274 10.9   68     5  14
```

Summary of missing values in R

```
n_complete(airquality)
```

```
## [1] 874
```

```
n_miss(airquality)
```

```
## [1] 44
```

Summary of missing values in R

```
miss_var_summary(airquality)
```

```
## # A tibble: 6 x 3
##   variable n_miss pct_miss
##   <chr>     <int>    <num>
## 1 Ozone        37    24.2
## 2 Solar.R       7     4.58
## 3 Wind          0     0
## 4 Temp          0     0
## 5 Month         0     0
## 6 Day           0     0
```

Summary of missing values in R

```
miss_case_summary(airquality)
```

```
## # A tibble: 153 x 3
##      case n_miss pct_miss
##     <int>  <int>    <dbl>
##  1     5      2     33.3
##  2    27      2     33.3
##  3     6      1     16.7
##  4    10      1     16.7
##  5    11      1     16.7
##  6    25      1     16.7
##  7    26      1     16.7
##  8    32      1     16.7
##  9    33      1     16.7
## 10    34      1     16.7
## # i 143 more rows
```

Summary of missing values in R

```
miss_var_table(airquality)
```

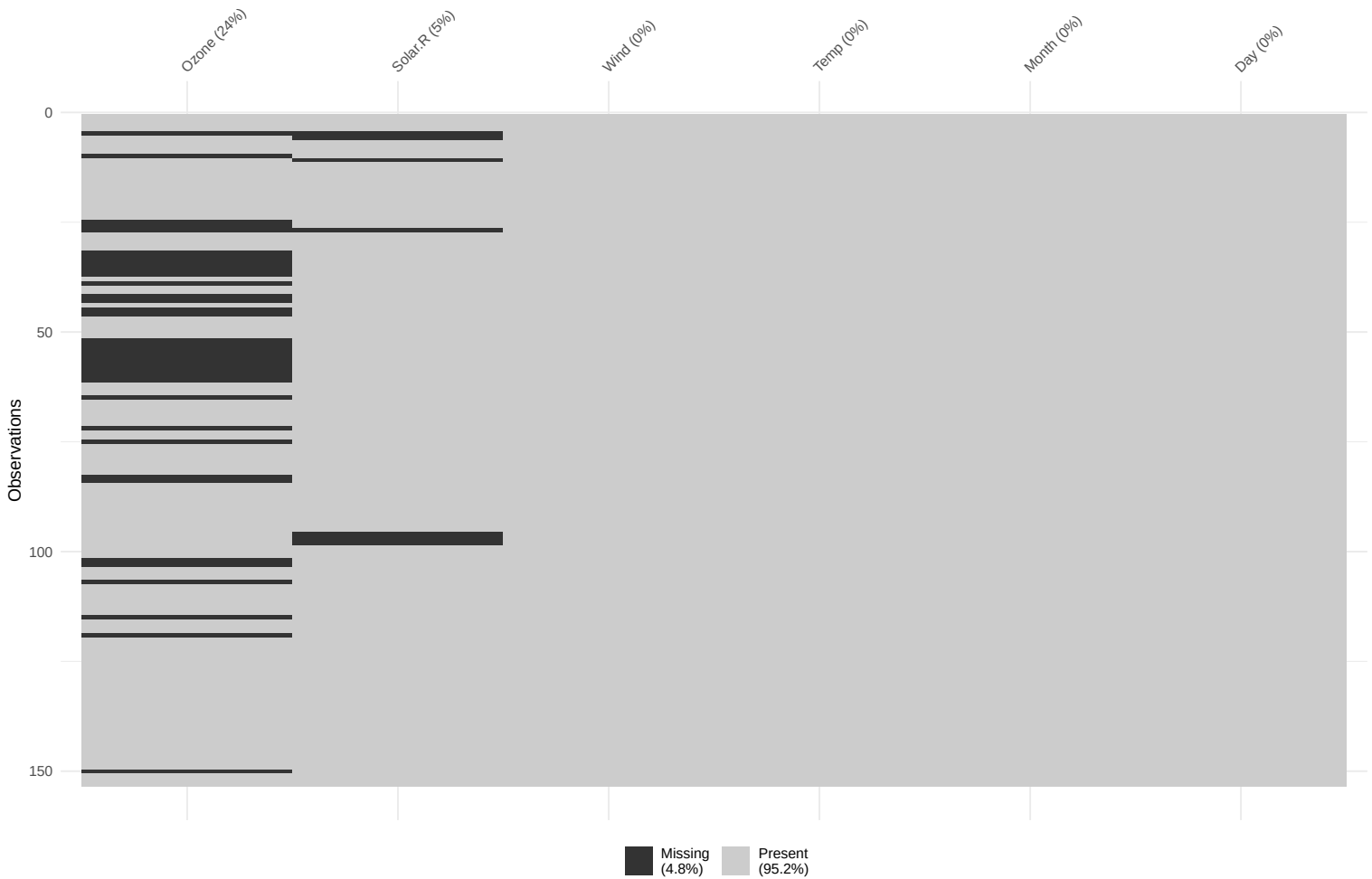
```
## # A tibble: 3 x 3
##   n_miss_in_var n_vars pct_vars
##           <int>  <int>    <dbl>
## 1             0      4     66.7
## 2             7      1     16.7
## 3            37      1     16.7
```

```
miss_case_table(airquality)
```

```
## # A tibble: 3 x 3
##   n_miss_in_case n_cases pct_cases
##            <int>   <int>     <dbl>
## 1              0     111     72.5
## 2              1      40     26.1
## 3              2       2      1.31
```

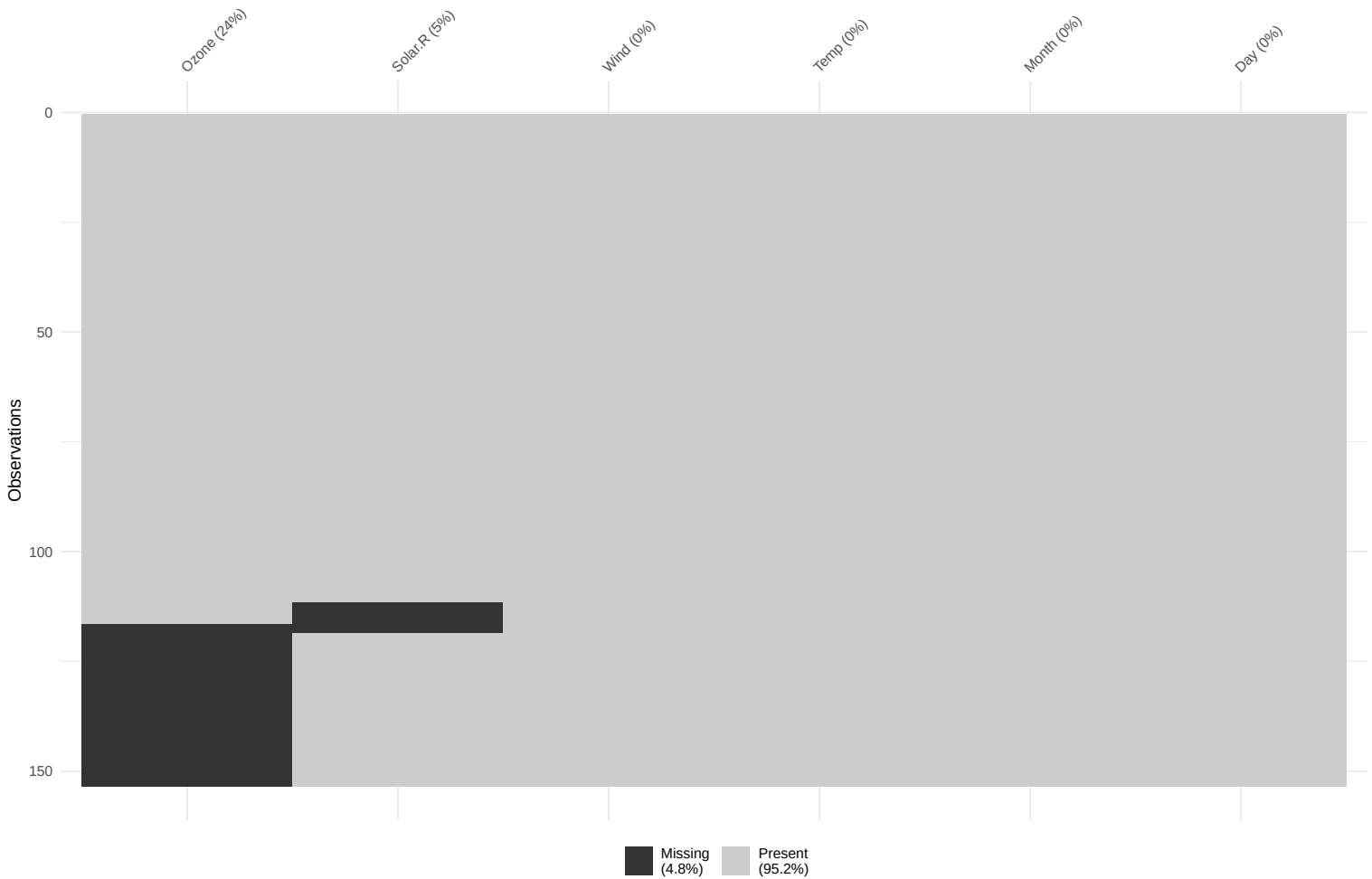
Visualizations of missing values in R

```
vis_miss(airquality)
```



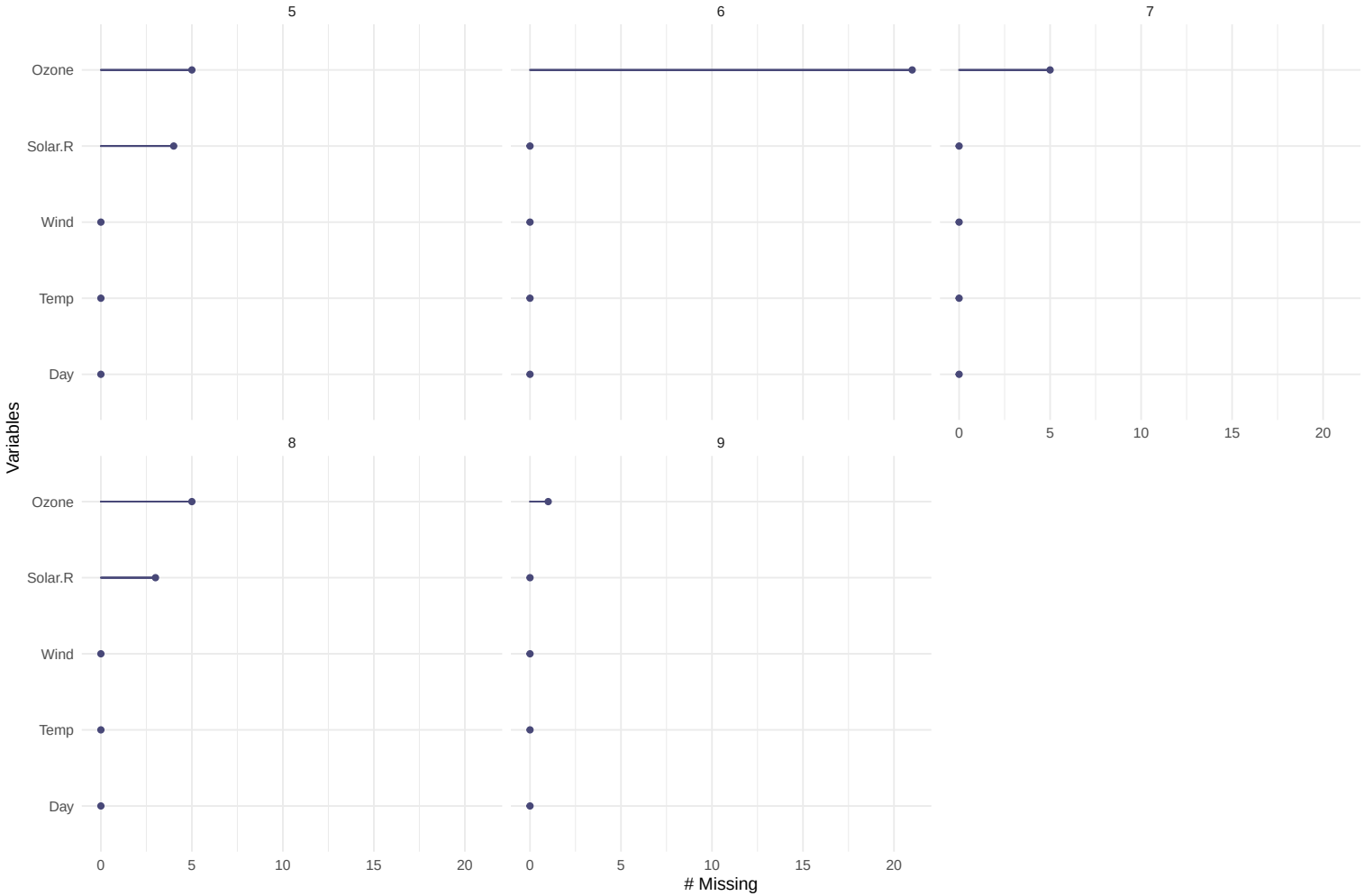
Visualizations of missing values in R

```
vis_miss(airquality,cluster = T)
```



Visualizations of missing values in R

```
gg_miss_var(airquality, facet = Month)
```



Dealing with Missing Values in practice

- No perfect way to deal with. It has to be case by case.
- Three ways to deal with
  - *deleting the observations*
  - *removing the variable*
  - *imputating*

Missing Values: Deleting the observations.

- The share of the missing observations is **minor**,let's say below 5%.
- or **"MCAR"** missing
- Then it can be dropped directly

```
df %>% drop_na()
```

```
## # A tibble: 6 x 5
##   customerID MonthlyCharges TotalCharges PaymentMethod    C
##   <chr>                <dbl> <chr>        <chr>            <
## 1 7590-VHVEG            29.8 109.9        Electronic check y
## 2 5575-GNVDE            57.0 na           Mailed check     y
## 3 7795-CFOCW            42.3 1840.75      Bank transfer    n
## 4 1452-KIOVK            89.1 1949.4       Credit card      n
## 5 7892-POOKP           105.  3046.05      Electronic check n
## 6 8451-AJOMK            54.1 354.95       Electronic check n
```

Missing Values: imputating

- What if
  1. deleting observations will leads a small sample size.
  2. the share of the missing observations is over 20%.
- **Imputating with statistics**
  - mean,median,mode
  - the next(previous) value
  - regression prediction or matching

Missing Values: imputating

- `tidyr::replace_na`: turn missing values(NAs) into specific values

```
df$PaymentMethod
```

```
##  [1] "Electronic check" "Mailed check"     "--"
##  [5] "Electronic check" "--"               "Credit card"
##  [9] "Electronic check" "Electronic check"
```

Replacing missing values

```
df %>%
replace_with_na(replace = list(PaymentMethod=c("","--"))) %>%
replace_na(list(PaymentMethod="Unknown"))
```

```
## # A tibble: 10 x 5
##    customerID MonthlyCharges TotalCharges PaymentMethod    Churn
##    <chr>                <dbl> <chr>        <chr>            <chr>
##  1 7590-VHVEG            29.8 109.9        Electronic check yes
##  2 5575-GNVDE            57.0 na           Mailed check     yes
##  3 3668-QPYBK             NA  108.15       Unknown          yes
##  4 7795-CFOCW            42.3 1840.75       Bank transfer    no
##  5 9237-HQITU            70.7 <NA>         Electronic check no
##  6 9305-CDSKC           NaN   820.5        Unknown          yes
##  7 1452-KIOVK            89.1 1949.4        Credit card      no
##  8 6713-OKOMC             NA  N/A          Unknown          yes
##  9 7892-POOKP           105.  3046.05       Electronic check no
## 10 8451-AJOMK            54.1 354.95       Electronic check no
```

Replacing missing values with statistics

```
df %>%
  mutate(MonthlyCharges=
           replace(MonthlyCharges,is.na(MonthlyCharges),
                   mean(MonthlyCharges, na.rm = TRUE)))
```

```
## # A tibble: 10 x 5
##    customerID MonthlyCharges TotalCharges PaymentMethod    Churn
##    <chr>               <dbl> <chr>        <chr>            <chr>
##  1 7590-VHVEG           29.8 109.9        Electronic check yes
##  2 5575-GNVDE           57.0 na           Mailed check     yes
##  3 3668-QPYBK           64.0 108.15       --               yes
##  4 7795-CFOCW           42.3 1840.75      Bank transfer    no
##  5 9237-HQITU           70.7 <NA>         Electronic check no
##  6 9305-CDSKC           64.0 820.5        --               yes
##  7 1452-KIOVK           89.1 1949.4       Credit card      no
##  8 6713-OKOMC           64.0 N/A          <NA>             yes
##  9 7892-POOKP          105.  3046.05      Electronic check no
## 10 8451-AJOMK           54.1 354.95       Electronic check no
```

Replacing missing values with statistics

- Exercise: Replacing missing values in `TotalCharges` with median values

Replacing missing values with statistics

- Exercise: Replacing missing values in TotalCharges with median values

```
df %>%
replace_with_na(replace = list(TotalCharges=c("na","N/A"))) %>%
  mutate(TotalCharges=as.numeric(TotalCharges)) %>%
  mutate(TotalCharges=replace(TotalCharges,is.na(TotalCharges),
                              median(TotalCharges,na.rm = TRUE)))
```

```
## # A tibble: 10 x 5
##    customerID MonthlyCharges TotalCharges PaymentMethod    Churn
##    <chr>               <dbl>        <dbl> <chr>            <chr>
##  1 7590-VHVEG           29.8         110. Electronic check yes
##  2 5575-GNVDE           57.0         820. Mailed check     yes
##  3 3668-QPYBK           NA           108. --               yes
##  4 7795-CFOCW           42.3        1841. Bank transfer    no
##  5 9237-HQITU           70.7         820. Electronic check no
##  6 9305-CDSKC          NaN           820. --               yes
##  7 1452-KIOVK           89.1        1949. Credit card      no
```

Section 3

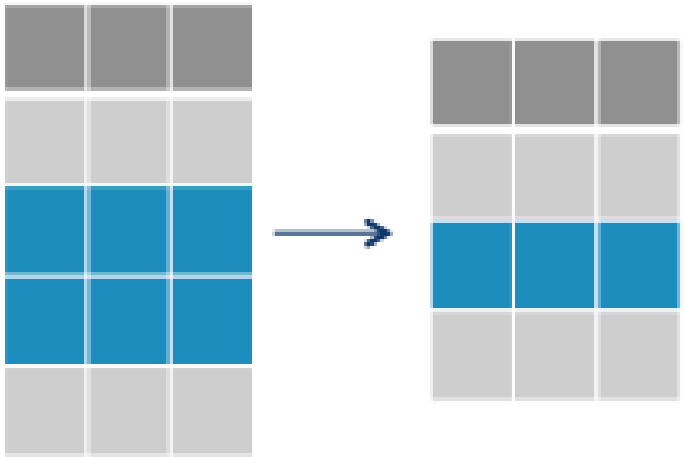
Duplicates

What is Duplicates?

- It means that
  - some observations are duplicated (maybe due to replicating records) in *all variables*.
  - some observation are only duplicated *only in one or several variables*.

Packages for Duplicates

Remove Duplicate Data in R



duplicated(): *Identify duplicate elements (R base)*
unique(): *Keep only unique elements (R base)*
distinct(): *Efficient solution to remove duplicate in a data table (dplyr)*

Find and drop duplicate elements

```
x <- c(1, 1, 4, 5, 4, 6)
duplicated(x)
```

```
## [1] FALSE  TRUE FALSE FALSE  TRUE FALSE
```

```
x[duplicated(x)] # Extract duplicate elements
```

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

```
x[!duplicated(x)] #remove duplicated elements
```

```
## [1] 1 4 5 6
```

Find and drop duplicate elements

```
unique(x)
```

```
## [1] 1 4 5 6
```

Excercise: Duplicates in R

```
# Create a sample dataset with duplicates
library(tibble)
duplicate_data <- tibble(
  id = c(1, 2, 2, 3, 3, 3, 4, 5, 5, 6),
  name = c("Alice", "Bob", "Bob", "Charlie", "Charlie", "Charlie", "David", "Eve",
  score = c(85, 92, 92, 78, 78, 78, 95, 88, 88, 91)
)

# Display original data
print(duplicate_data)
```

```
## # A tibble: 10 x 3
##       id name    score
##    <dbl> <chr>   <dbl>
##  1     1 Alice      85
##  2     2 Bob        92
##  3     2 Bob        92
##  4     3 Charlie    78
##  5     3 Charlie    78
##  6     3 Charlie    78
##  7     4 David      95
##  8     5 Eve        88
##  9     5 Eve        88
## 10     6 Frank      91
```

Excercise: Duplicates in R

```
# Check for duplicate rows using duplicated()

print(duplicated(duplicate_data))
```

```
##  [1] FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE  TRUE
```

```
# Display duplicate rows

print(duplicate_data[duplicated(duplicate_data), ])
```

```
## # A tibble: 4 x 3
##      id name    score
##   <dbl> <chr>   <dbl>
## 1     2 Bob        92
## 2     3 Charlie    78
## 3     3 Charlie    78
## 4     5 Eve        88
```

Section 4

Label Variables

Introduction

- Variable label is human readable description of the variable.
  - Labeling Variables
  - Labeling Values of Variables

Excercise: label for mtcars

```
library(expss)
data("mtcars")
mtcars
```

```
##                      mpg cyl  disp  hp drat    wt  qsec vs am gea
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0
```

Label Variables

```
apply_labels(mtcars,
                      mpg = "Miles/(US) gallon",
                      cyl = "Number of cylinders",
                      disp = "Displacement (cu.in.)",
                      hp = "Gross horsepower"
)
```

```
##                    mpg cyl  disp  hp drat    wt  qsec vs am gea
## Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1
## Mazda RX4 Wag     21.0   6 160.0 110 3.90 2.875 17.02  0  1
## Datsun 710        22.8   4 108.0  93 3.85 2.320 18.61  1  1
## Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0
## Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0
## Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0
## Duster 360        14.3   8 360.0 245 3.21 3.570 15.84  0  0
## Merc 240D         24.4   4 146.7  62 3.69 3.190 20.00  1  0
## Merc 230          22.8   4 140.8  95 3.92 3.150 22.90  1  0
## Merc 280          19.2   6 167.6 123 3.92 3.440 18.30  1  0
## Merc 280C         17.8   6 167.6 123 3.92 3.440 18.90  1  0
```

Label Values of Variables

```
mtcars = apply_labels(mtcars,
                      vs = "Engine",
                      vs = num_lab("
                             0 V-engine
                             1 Straight engine
                             "),
                      am = "Transmission",
                      am = num_lab("
                             0 Automatic
                             1 Manual
                             ")
)
```

Tables: variables with labels

```
table(mtcars$vs, mtcars$am)
```

```
##
##                   Automatic Manual
##   V-engine                12      6
##   Straight engine          7      7
```

```
calculate(mtcars, cro(vs, am))
```

Transmission

Automatic

Manual

Engine

V-engine

12

Section 5

Making journal-quality tables in Rmarkdown

A Simple Table by hand in Markdown

Topic	Date	Time
HW1	10/13/2019	24:00pm

A Simple Table by hand in Markdown

Right	Left	Default	Center
123	123	123	123

Useful packages

- Packages
  - `xtable`
  - `kable` and `kableExtra`
  - `stargazer`

Read Data into R

```
library(haven)
rand <- read_dta("Data/rand.dta")
structure(rand)
```

```
## # A tibble: 5,887 x 10
##    educper  hosp female cholest cholestx blackhisp ghindx ghindxx  famid
##      <dbl> <dbl>  <dbl>   <dbl>    <dbl>     <dbl>  <dbl>   <dbl>  <dbl>
##  1      NA    NA     NA      NA       NA        NA     NA      NA     NA
##  2      12     0      0      NA      245         1     NA    71.6 100082
##  3      NA     0      0      NA      207        NA     NA    69.3 100082
##  4      NA     0      1      NA      161        NA     NA    92   100082
##  5      12     0      1      NA      176         1     NA    73.9 100082
##  6      NA    NA     NA      NA       NA        NA     NA    75     NA
##  7      NA     0      1     212      185        NA     NA    65.9 100240
##  8      12     0      1     328      324         1     NA    44.3 100240
##  9      11     0      0     192       NA        NA     NA    NA   100240
## 10      12     0      1     170       NA         1     NA    NA   100259
## # i 5,877 more rows
```

Install Packages

```
#install.packages("kableExtra")
#install.packages("stargazer")
library(kableExtra)
library(stargazer)
```

Summary Statistics

```
ra <- rand %>% drop_na() %>%
  group_by(any_ins) %>%
  summarise("mean(female)" = mean(female, na.rm=T), "sd(female
            "mean(blackhsip)" = mean(blackhisp, na.rm=T),"sd(b
            "mean(educper)" = mean(educper, na.rm=T), "sd(educ
            "mean(ghindx)" = mean(educper, na.rm=T), "sd(ghind
            "mean(cholest)" = mean(educper, na.rm=T), "sd(chol
print(ra)
```

```
## # A tibble: 2 x 12
##   any_ins `mean(female)` `sd(female)` `mean(blackhsip)` `sd
##     <dbl>          <dbl>        <dbl>              <dbl>
## 1       0          0.565        0.497              0.175
## 2       1          0.545        0.498              0.140
## # i 7 more variables: `mean(educper)` <dbl>, `sd(educper)`
## #   `mean(ghindx)` <dbl>, `sd(ghindx)` <dbl>, `mean(cholest
```

Summary Statistics

```
kable(ra,"latex",)
```

any_ins	mean(female)	sd(female)	mean(blackhsip)	sd(blackhisp)	r
0	0.5650224	0.4968694	0.1748879	0.3807258	
1	0.5448889	0.4982024	0.1404444	0.3476021	

Summary Statistics

```
ra %>% mutate_if(is.numeric,format,digits=3) %>%
kable(., "html") %>%
  kable_styling(bootstrap_options = "striped",
                full_width = F,position ="center")
```

any_ins	mean(female)	sd(female)	mean(blackhsip)	sd(blackhisp)	mean(educper)	sd(educper)	mean(ghindx)
0	0.565	0.497	0.175	0.381	12.3	3.13	12.3
1	0.545	0.498	0.140	0.348	12.2	3.05	12.2

T.Test in R

```
g.ttest <- with(rand,t.test(ghindx ~ any_ins))
g.ttest
```

```
##
##  Welch Two Sample t-test
##
## data:  ghindx by any_ins
## t = 2.2577, df = 4037.9, p-value = 0.02402
## alternative hypothesis: true difference in means between gr
## 95 percent confidence interval:
##  0.1349045 1.9150177
## sample estimates:
## mean in group 0 mean in group 1
##        70.95892        69.93396
```

T.Test in R

```
c.ttest <- with(rand,t.test(cholest ~ any_ins))
```

T.Test in table

```
library(broom)
library(purrr)
tab <- map_df(list(g.ttest,c.ttest), tidy)%>%
  mutate_if(is.numeric,format,digits=3)
tab_less<-tab[c("estimate", "statistic",
                "p.value", "conf.low", "conf.high")]
kable(tab_less,"latex")
```

estimate	statistic	p.value	conf.low	conf.high
1.02	2.26	0.0240	0.135	1.92
2.98	1.78	0.0746	-0.296	6.25

Section 6

Rmarkdown's Chunk

Section 7

Data visualization: ggplot2 in Tidyverse

Introduction

- ggplot2 is a powerful data exploration and visualization package that can create graphics in R. It was created by Hadley Wickham who is a leading developer of R package.
- This function is the implementation of the "Grammar of Graphics" that allows us to build layers of graphical elements to produce plots.

```
library(ggplot2)
library(gapminder)
```

Terminology in ggplot2

- **ggplot** - the main function where you specify the data set and variables to plot (this is where we define the `x` and `y` variable names)
- **geoms** - geometric objects
  - e.g. `geom_point()`, `geom_bar()`, `geom_line()`, `geom_histogram()`,`geom_boxplot()`
- **aes** - aesthetics
  - shape, transparency, color, fill, linetype
- **scales** - define how your data will be plotted
  - continuous, discrete, log, etc

Scatter Plot

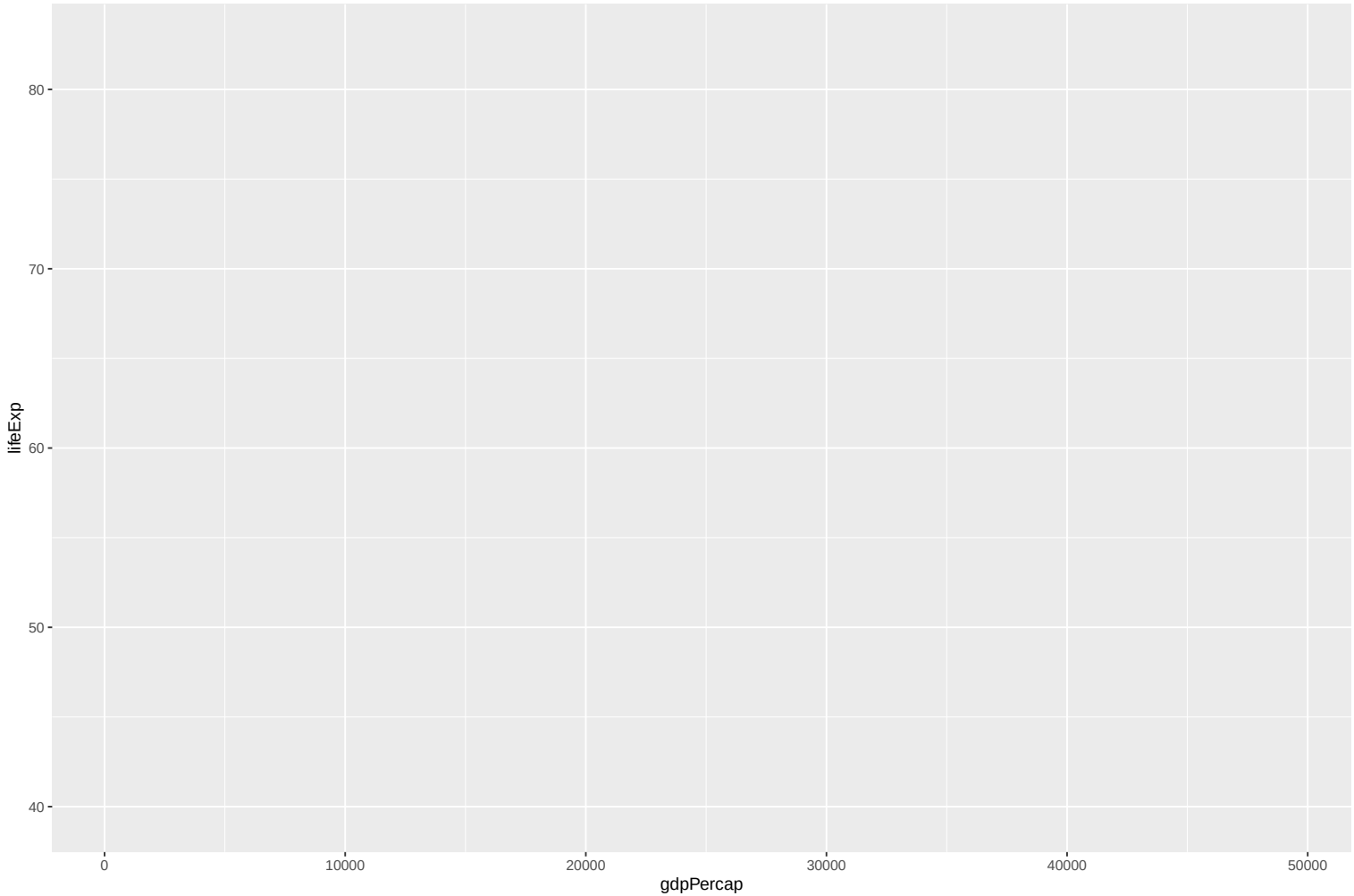
- generate a subset data in 2007

```
gapminder_2007 <- gapminder %>%
    filter(year==2007)
glimpse(gapminder_2007)
```

```
## Rows: 142
## Columns: 6
## $ country   <fct> "Afghanistan", "Albania", "Algeria", "Ang
## $ continent <fct> Asia, Europe, Africa, Africa, Americas, O
## $ year      <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007,
## $ lifeExp   <dbl> 43.828, 76.423, 72.301, 42.731, 75.320, 8
## $ pop       <int> 31889923, 3600523, 33333216, 12420476, 40
## $ gdpPercap <dbl> 974.5803, 5937.0295, 6223.3675, 4797.2313
```

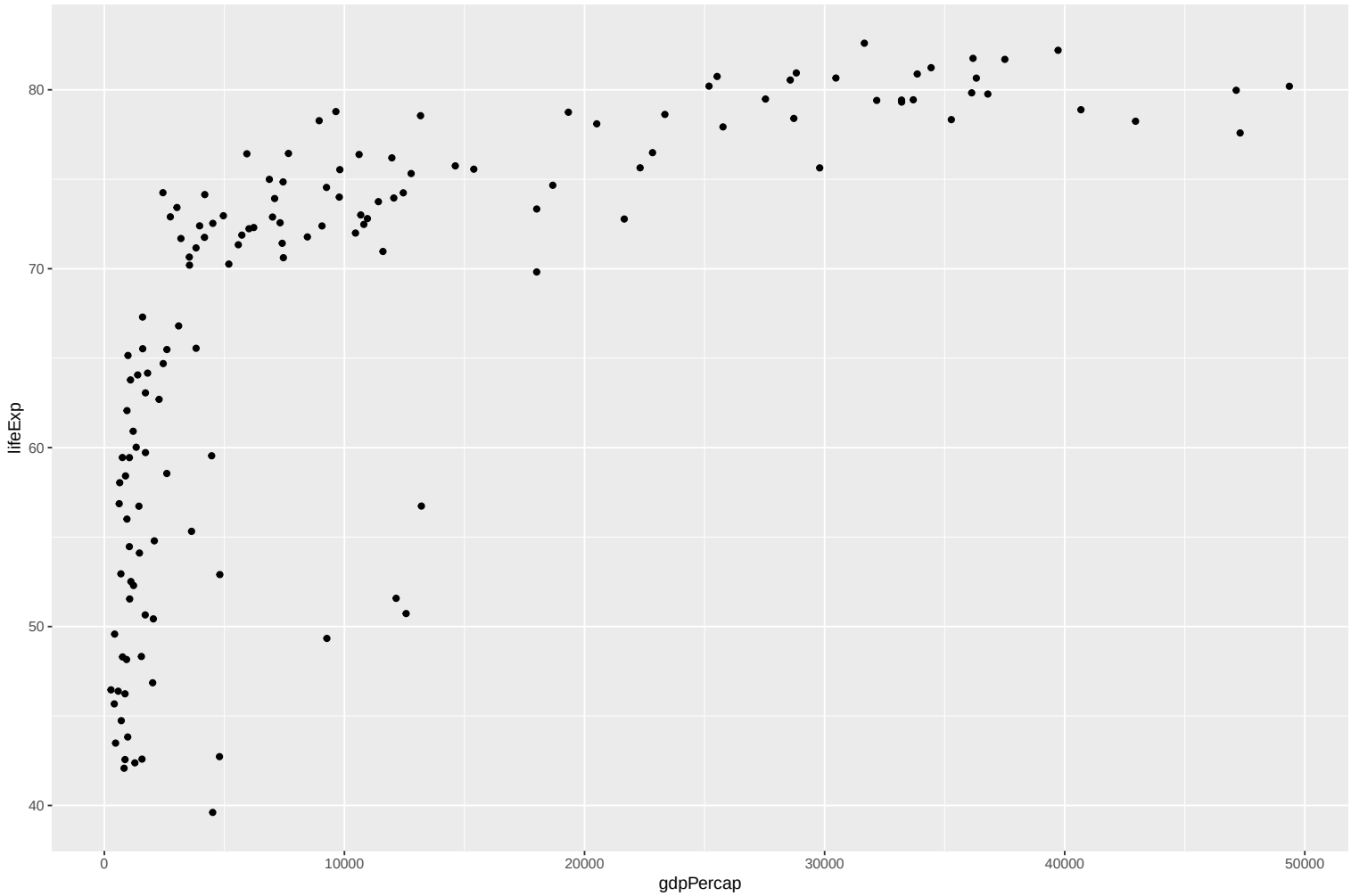
Scatter Plot

```
ggplot(gapminder_2007,aes(x=gdpPercap, y = lifeExp))
```



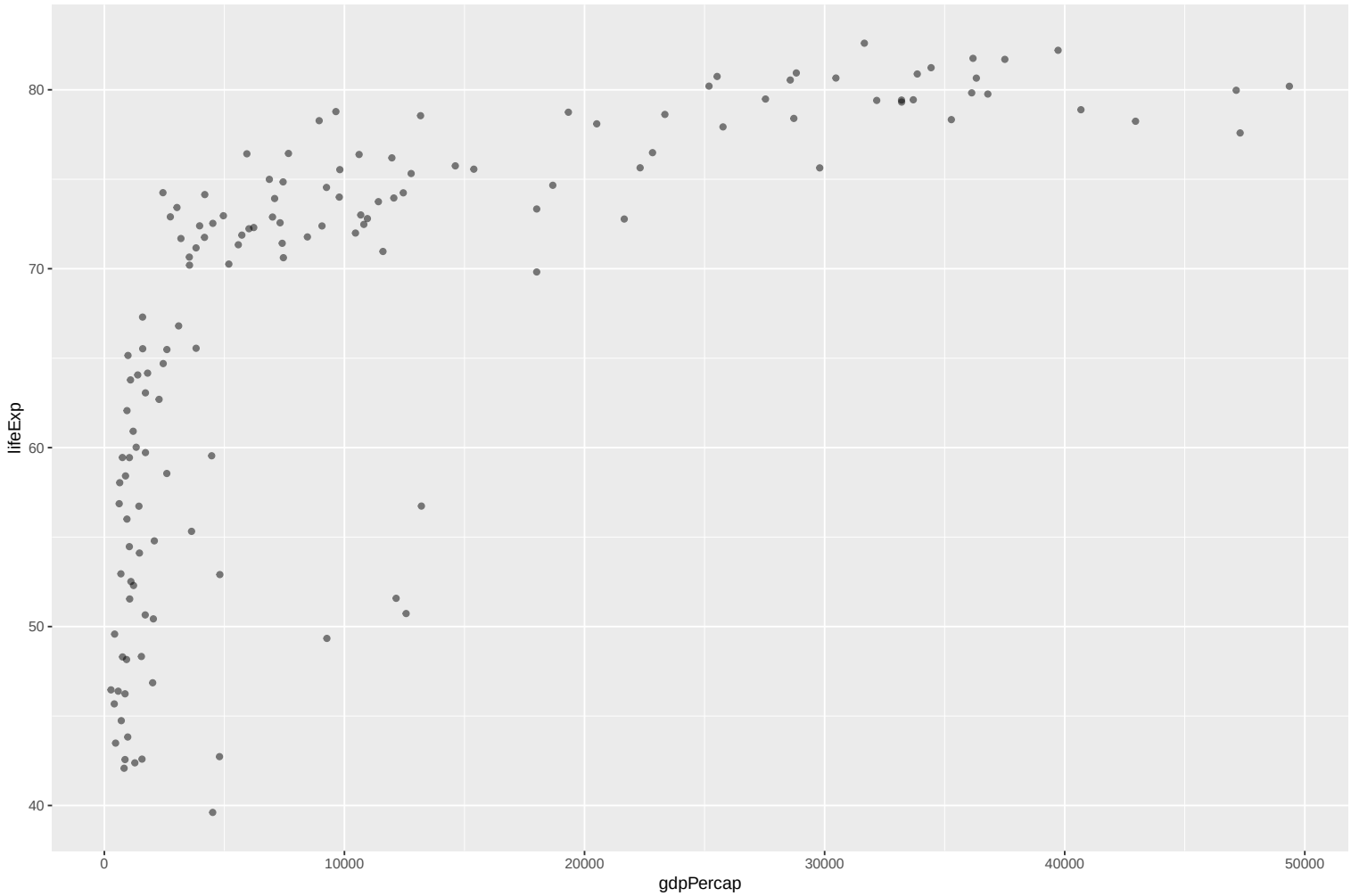
Scatter Plot

```
ggplot(gapminder_2007,aes(x=gdpPercap, y = lifeExp)) +
geom_point()
```



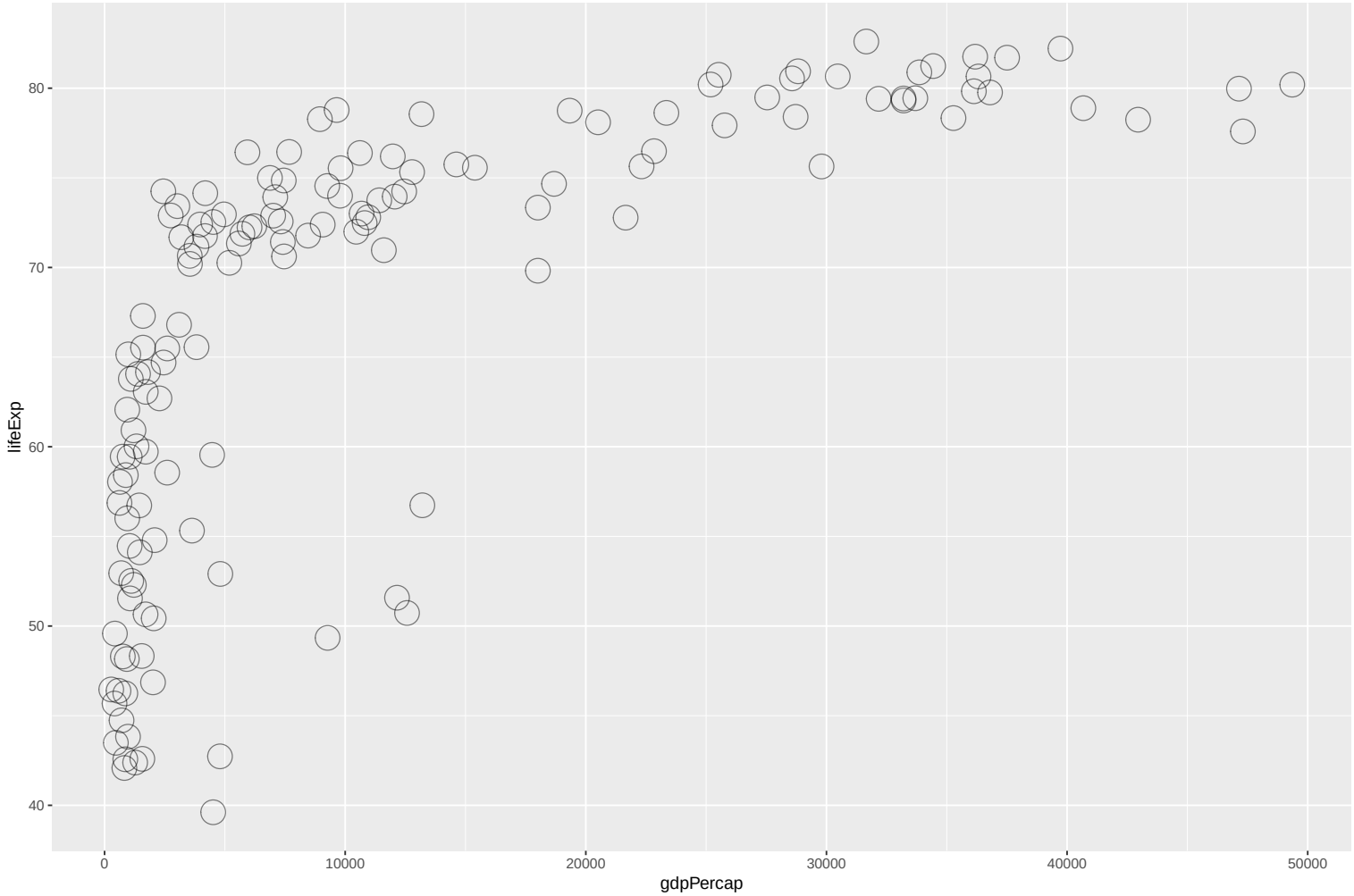
Scatter Plot

```
ggplot(gapminder_2007,aes(x=gdpPercap, y = lifeExp)) +
geom_point(alpha=0.5)
```



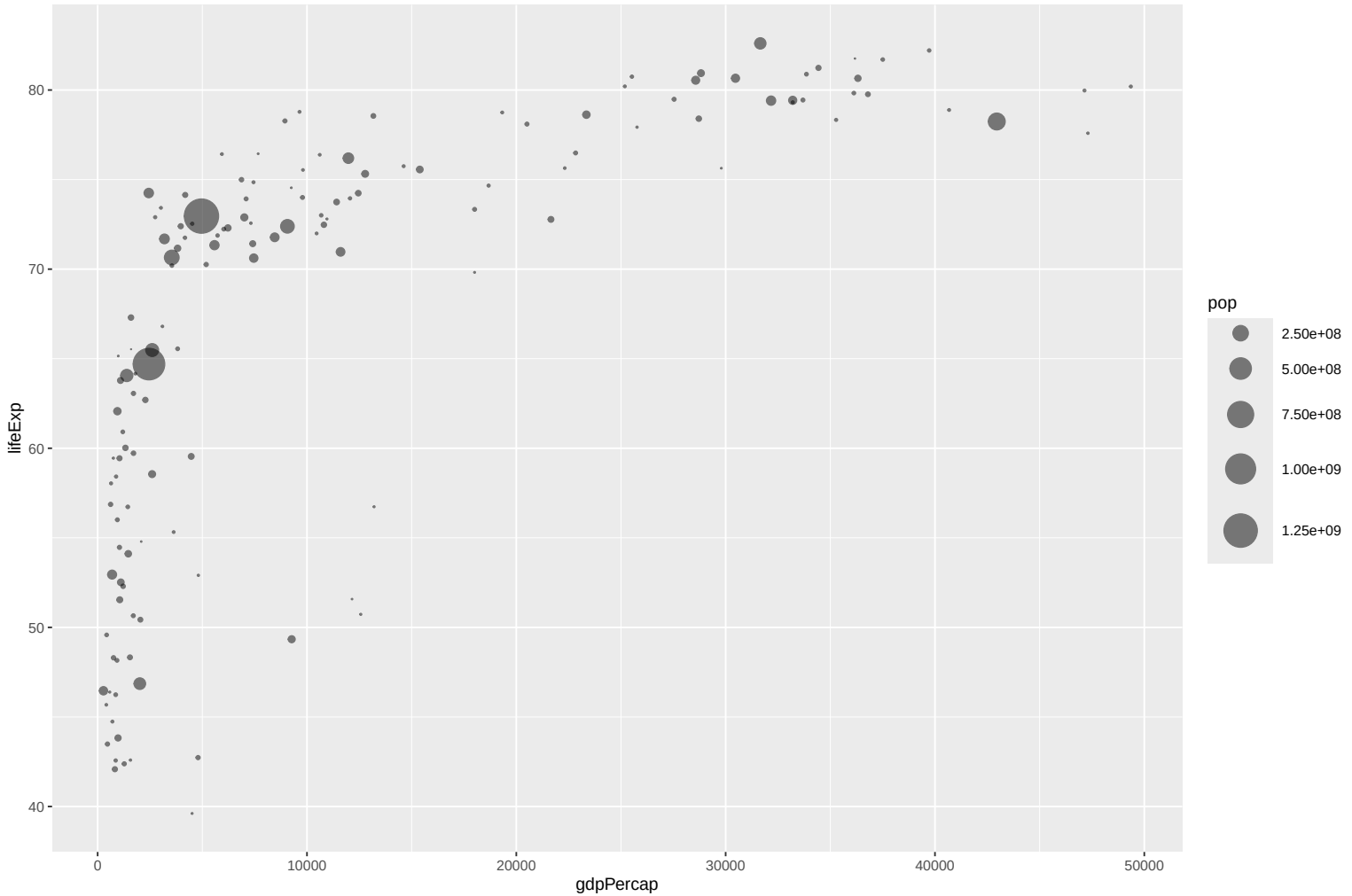
Scatter Plot

```
ggplot(gapminder_2007,aes(x=gdpPercap, y = lifeExp)) +
geom_point(alpha=0.5,shape=21,size=7)
```



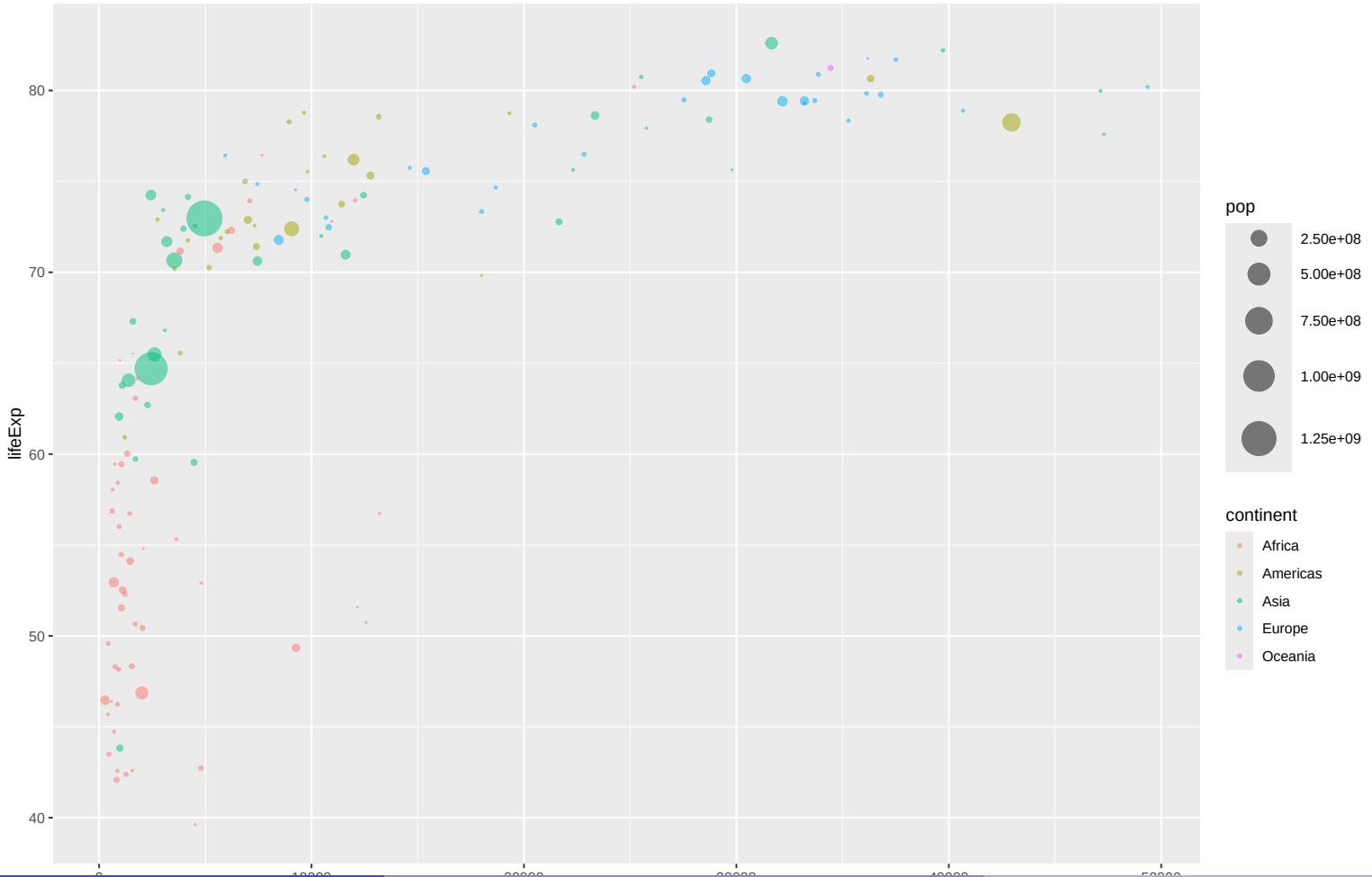
Scatter Plot

```
ggplot(gapminder_2007,aes(x=gdpPercap, y = lifeExp,size=pop)) +
geom_point(alpha=0.5,shape=20)+scale_size(range=c(.1,15))
```



Scatter Plot in colors

```
ggplot(gapminder_2007,
    aes(x=gdpPercap, y = lifeExp,size=pop,color=continent)) +
    geom_point(alpha=0.5,shape=20)+scale_size(range=c(.1,15))
```



Scatter Plot:label in ggrepel

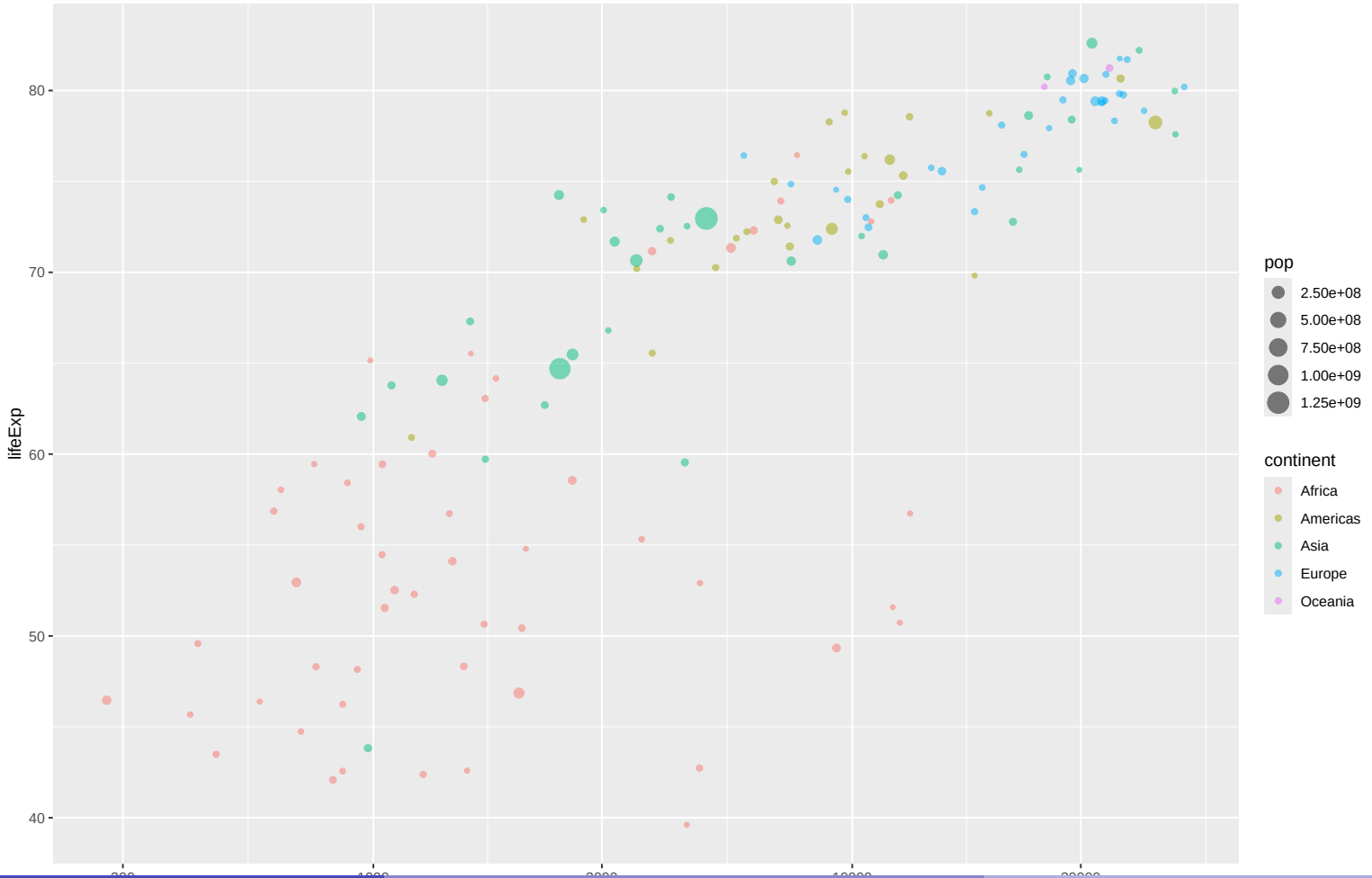
```
# install.packages("ggrepel")
library(ggrepel)
gapminder %>%
    filter(year==2007) %>%
    ggplot(aes(x=gdpPercap,
               y = lifeExp,size = pop,color=continent)) +
    geom_point(alpha=0.5)+scale_size(range=c(.1,15)) +
    geom_text_repel(aes(label = country), size = 2) +
    theme_classic()
```

Scatter Plot:label in ggrepel

```
## Warning: ggrepel: 6 unlabeled data points (too many overlaps). C
## increasing max.overlaps
```

Scatter Plot: Log X

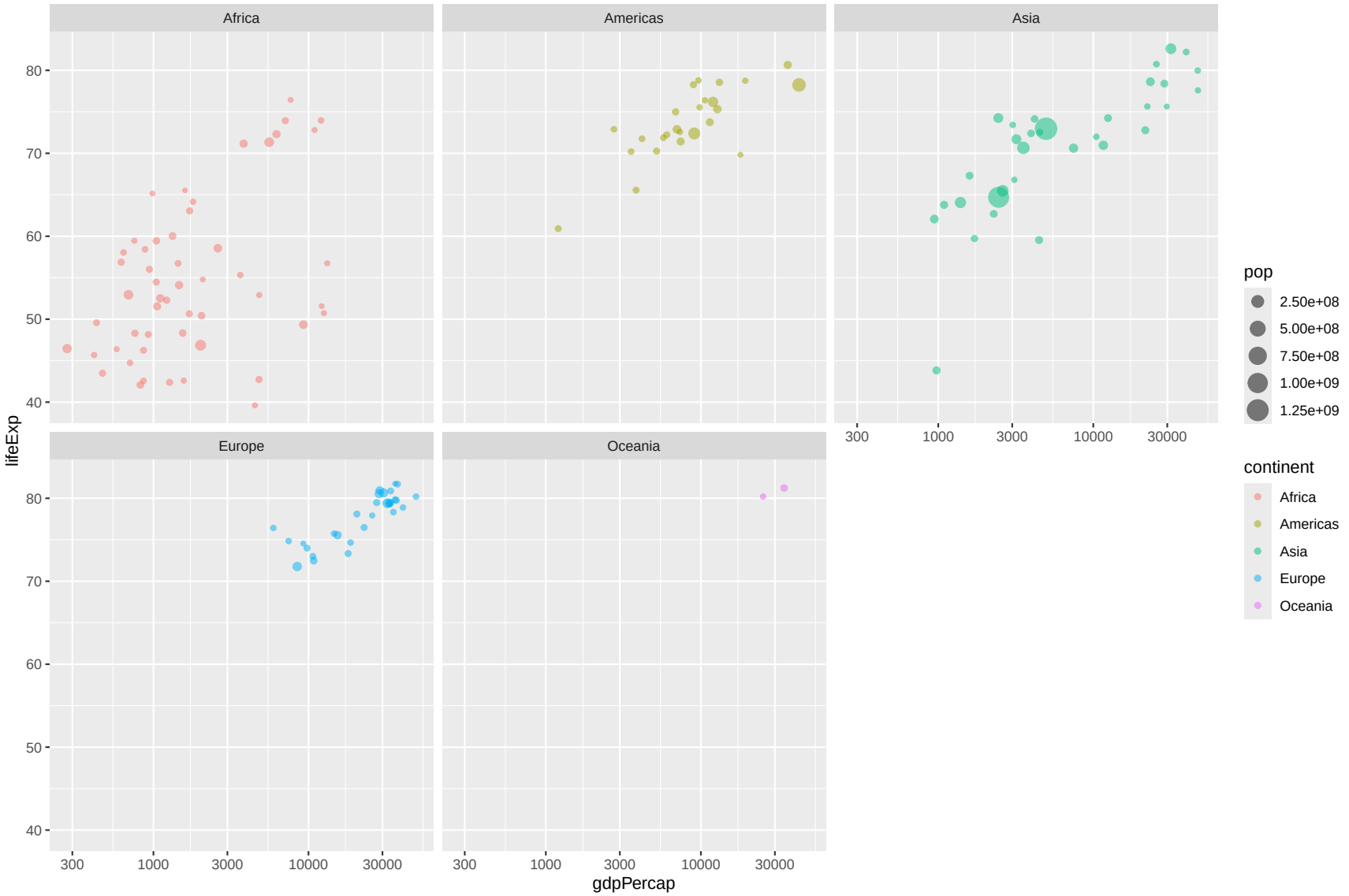
```
ggplot(gapminder_2007,
       aes(x=gdpPercap,y = lifeExp,size=pop,color=continent)) +
geom_point(alpha=0.5)+scale_x_log10()
```



Scatter Plot: by group Facet

```
ggplot(gapminder_2007,
       aes(x=gdpPercap, y = lifeExp,size = pop,
           color=continent)) +
geom_point(alpha=0.5) +
scale_x_log10()+
facet_wrap(~ continent)
```

Scatter Plot: by group Facet



Scatter Plot: by group Facet

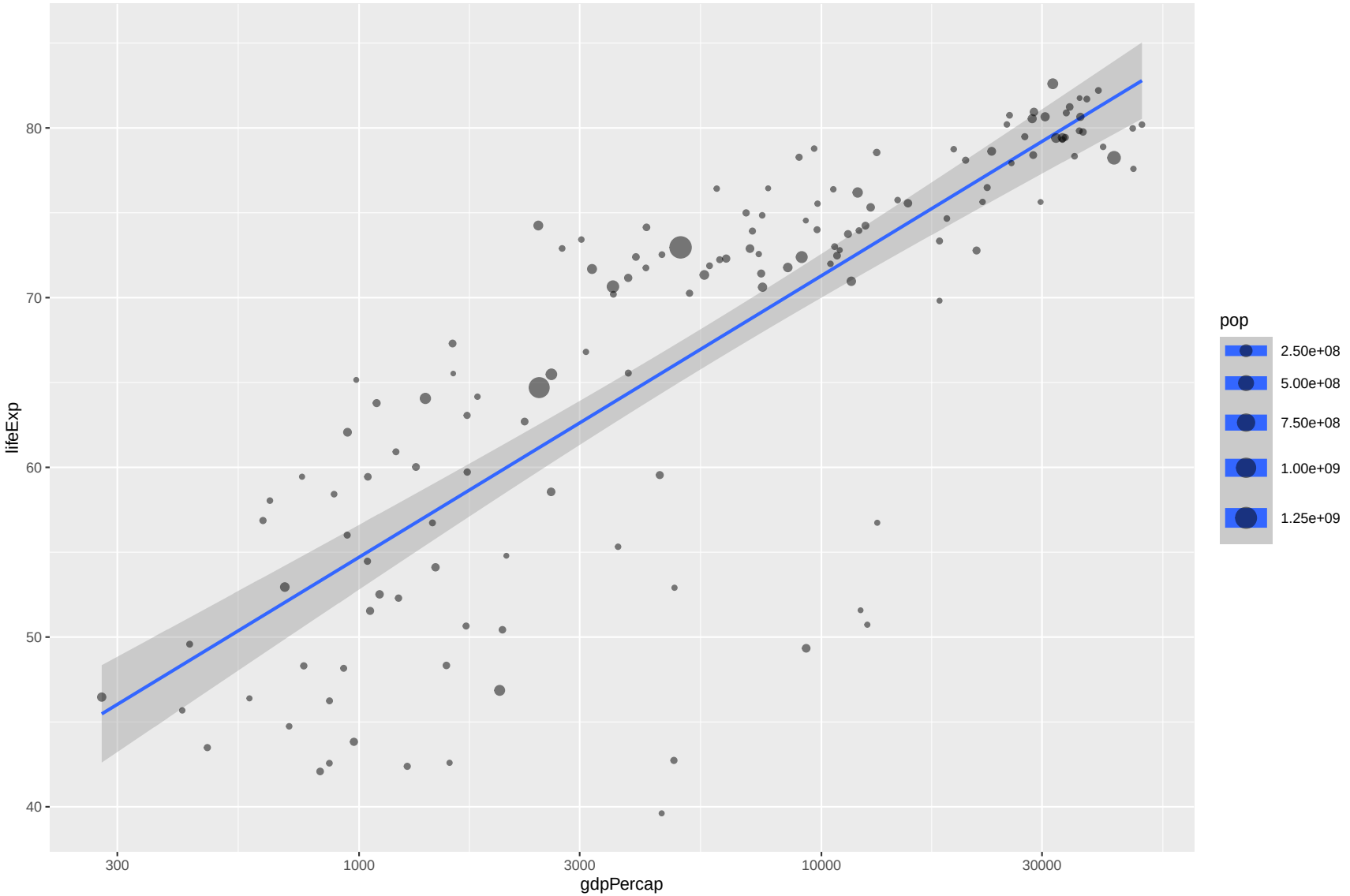
```
ggplot(gapminder,aes(x=gdpPercap, y = lifeExp,size = pop,
                     color = continent)) +
geom_point() +scale_x_log10()+facet_wrap(~ year)
```

Scatter Plot: by group Facet

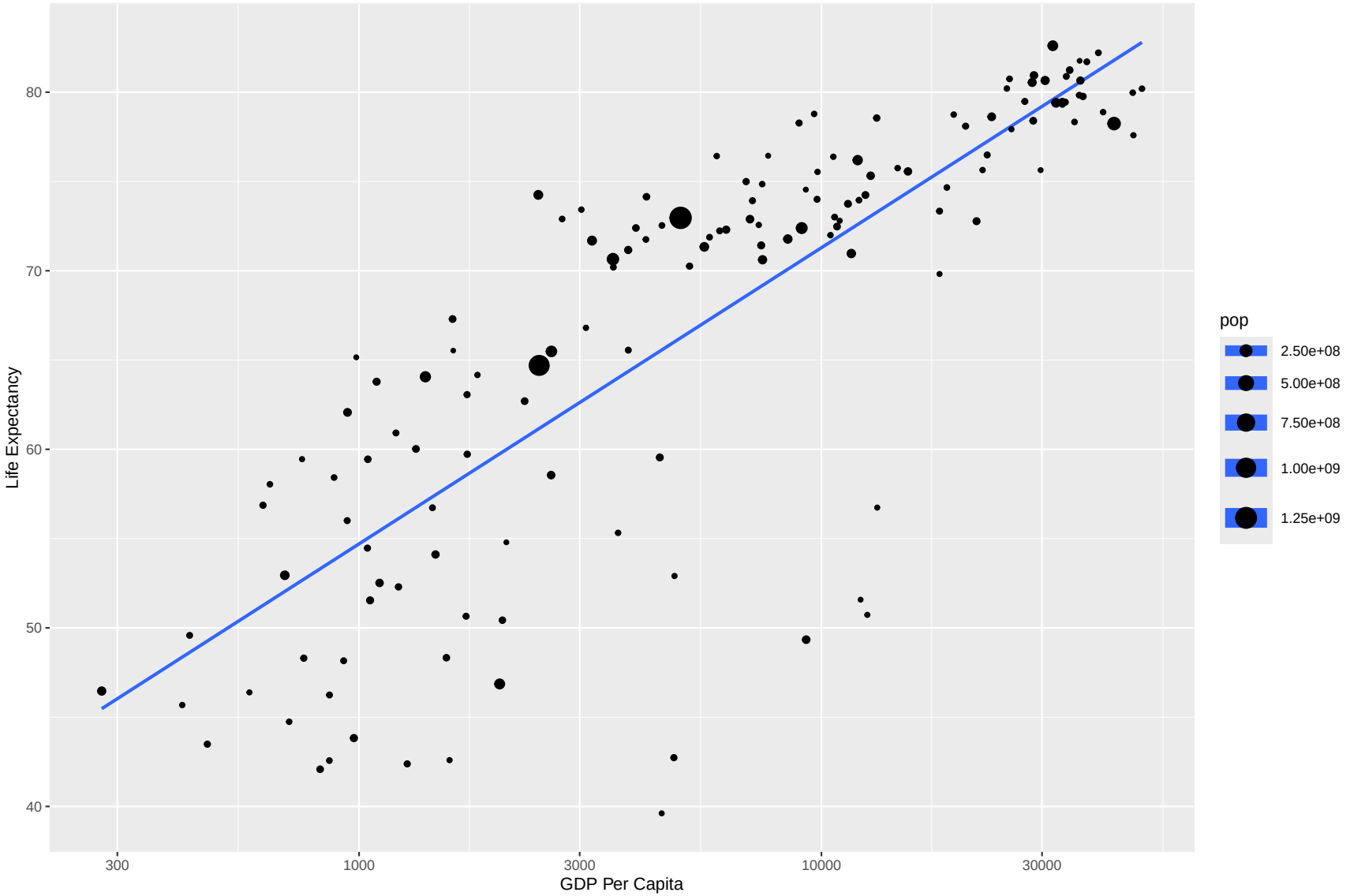
Scatter Plot: regression line

```
gapminder %>%
    filter(year==2007) %>%
    ggplot(aes(x=gdpPercap, y = lifeExp,size = pop)) +
    geom_smooth(method = lm) +
    geom_point(alpha=0.5) +
    scale_x_log10()
```

Scatter Plot: regression line

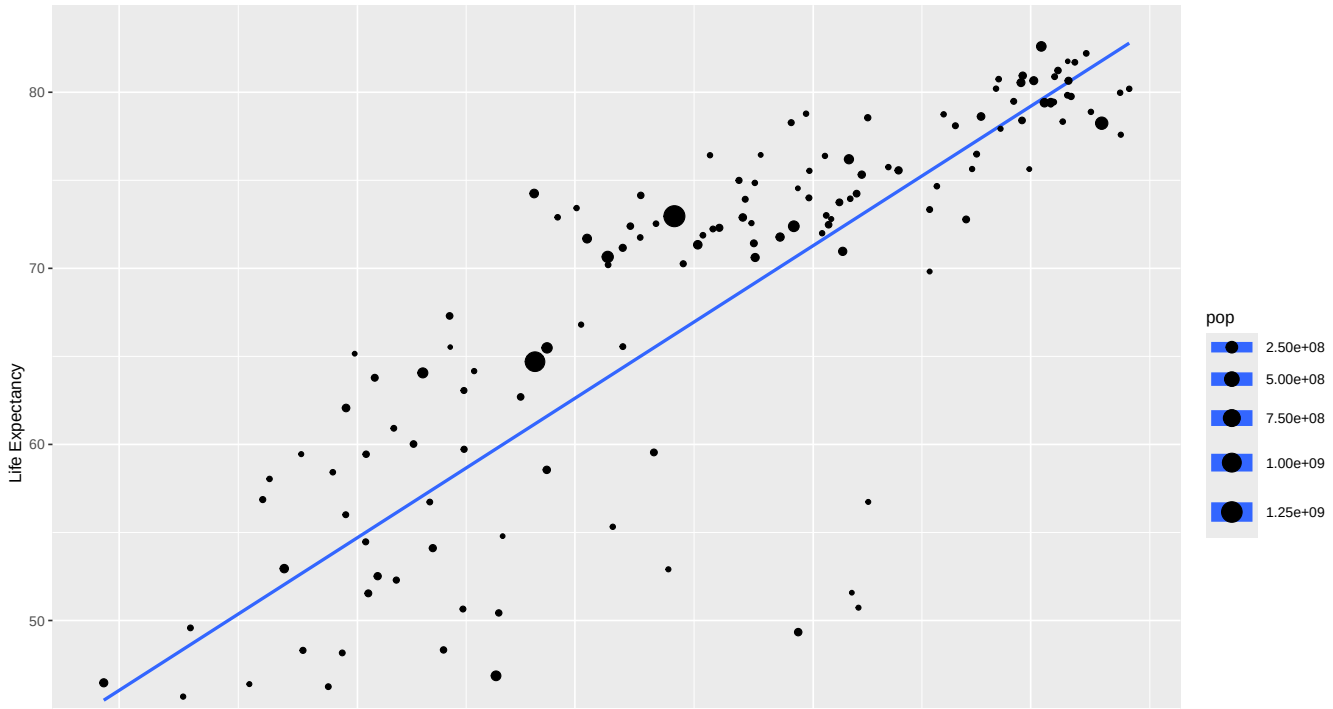


Scatter Plot: Straight line



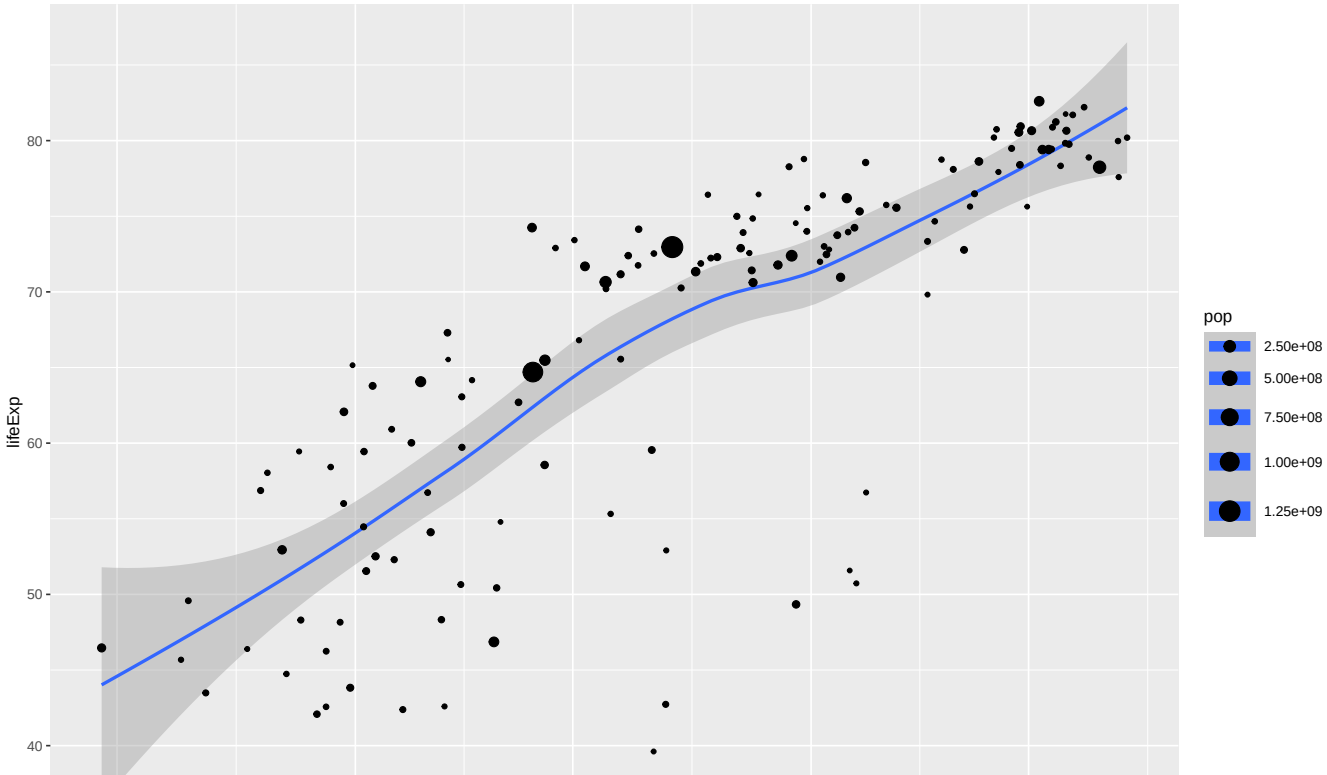
Scatter Plot:

```
gapminder %>%
    filter(year==2007) %>%
    ggplot(aes(x=gdpPercap, y = lifeExp,size = pop)) +
    geom_smooth(method = lm,se=FALSE) +
    geom_point() +
    scale_x_log10()+
    labs(x="GDP Per Capita",y="Life Expectancy")
```



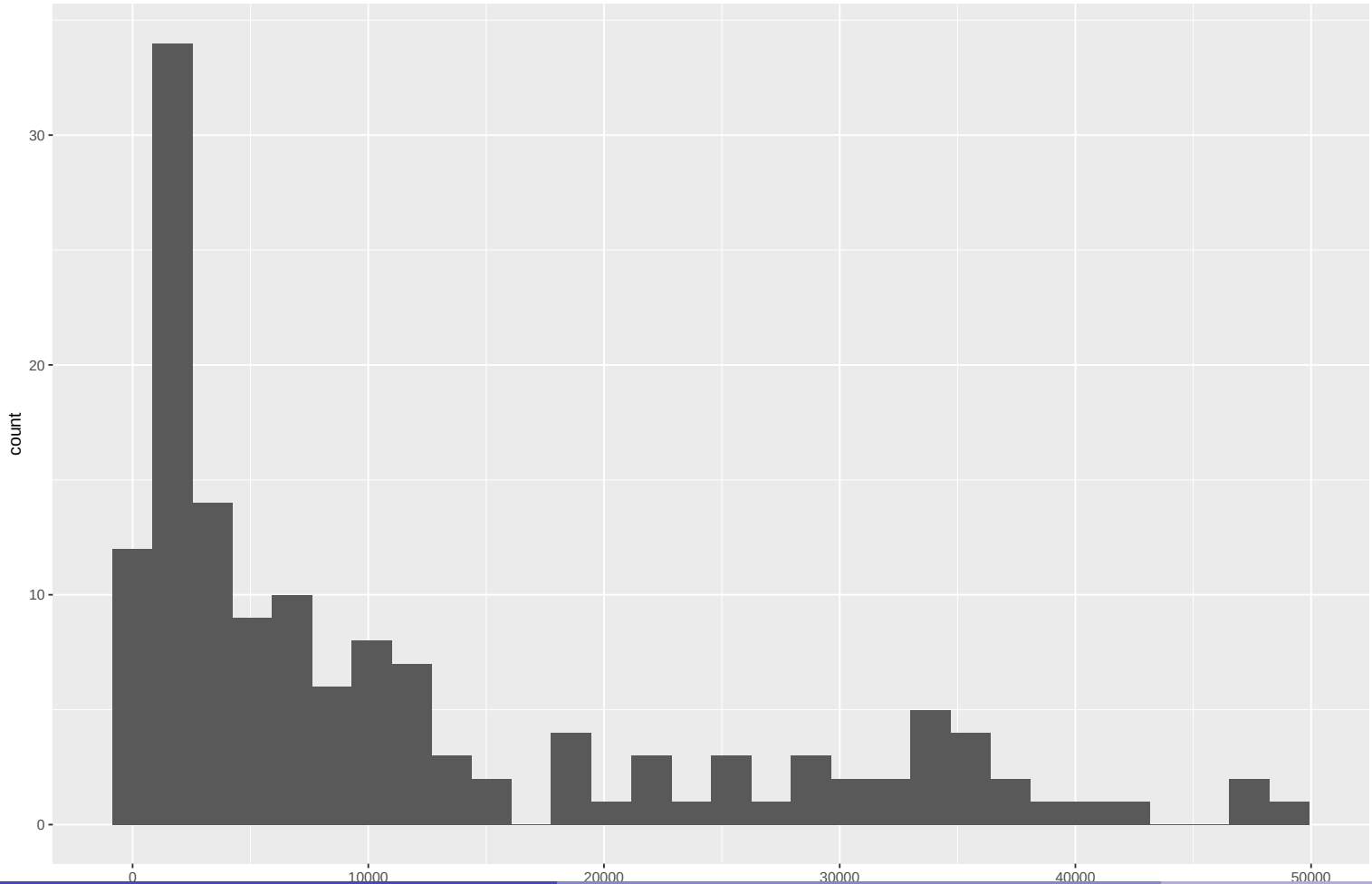
Scatter Plot: smoothing line

```
gapminder %>%
    filter(year==2007) %>%
    ggplot(aes(x=gdpPercap, y = lifeExp,size = pop)) +
    geom_smooth() +
    geom_point() +
    scale_x_log10()
```



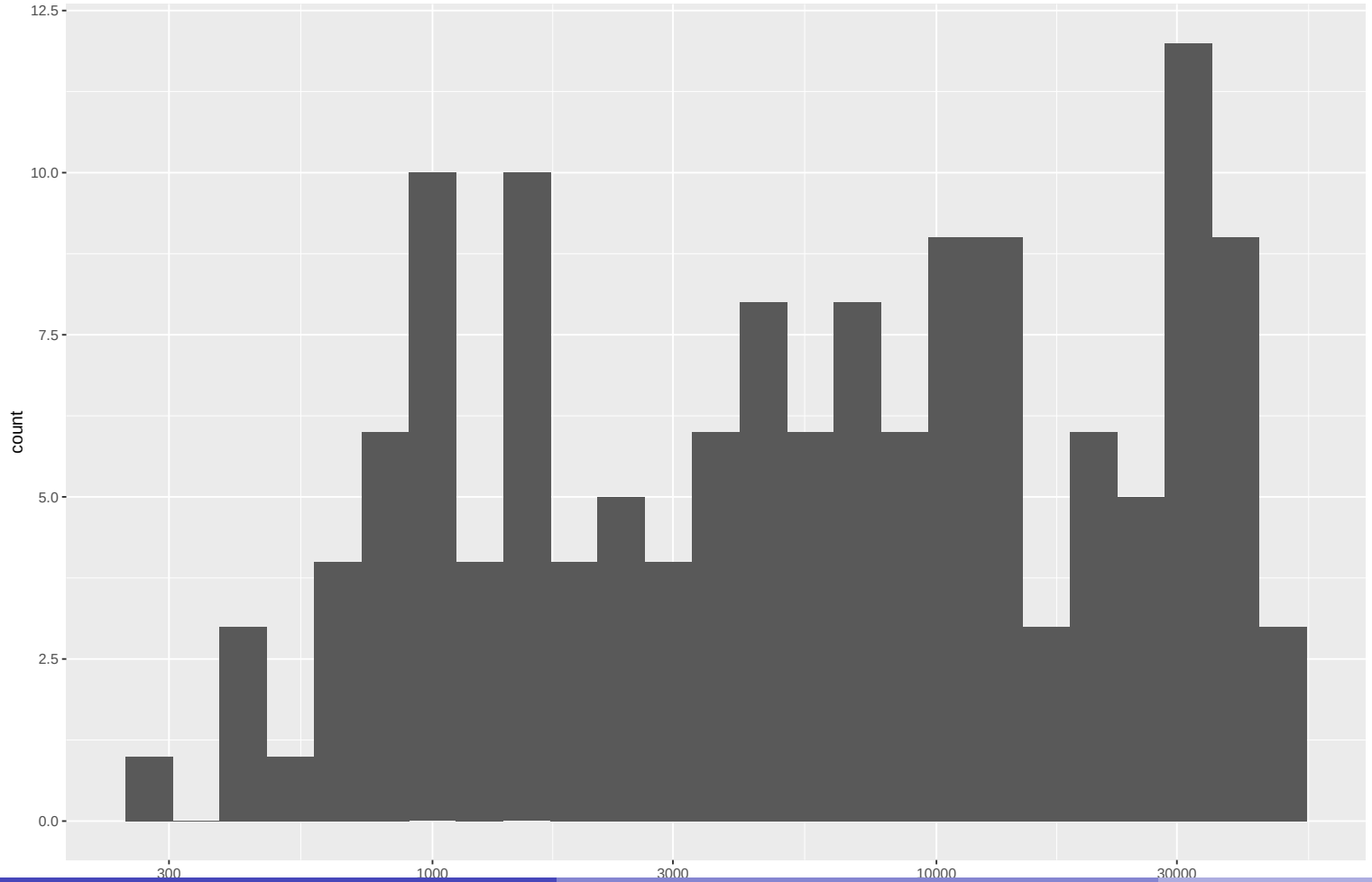
Histogram

```
gapminder %>%
    filter(year==2007) %>% ggplot(aes(x=gdpPercap))+
    geom_histogram()
```



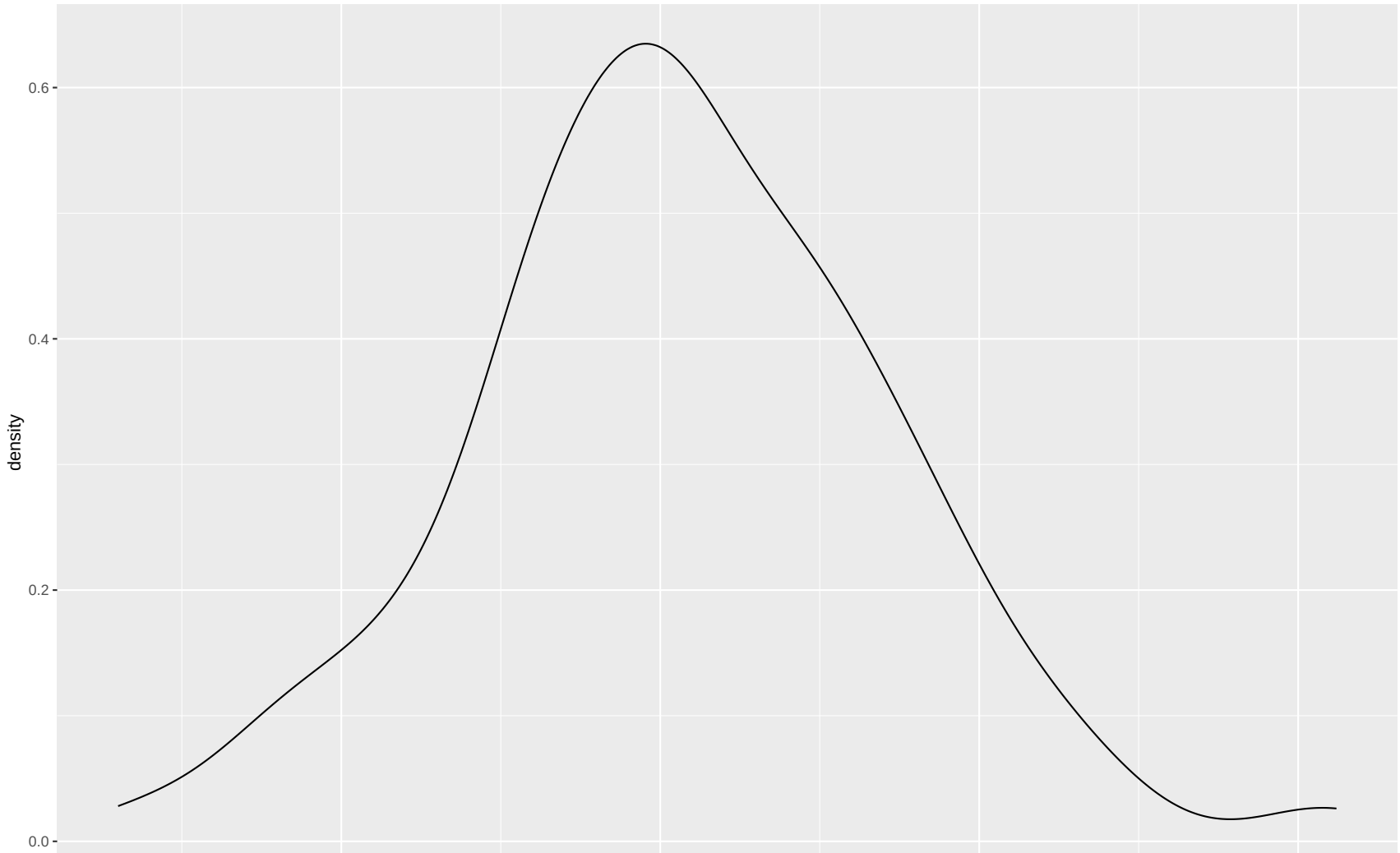
Histogram

```
gapminder %>%
    filter(year==2007) %>% ggplot(aes(x=gdpPercap))+
    geom_histogram(bins=25) + scale_x_log10()
```



Kdensity

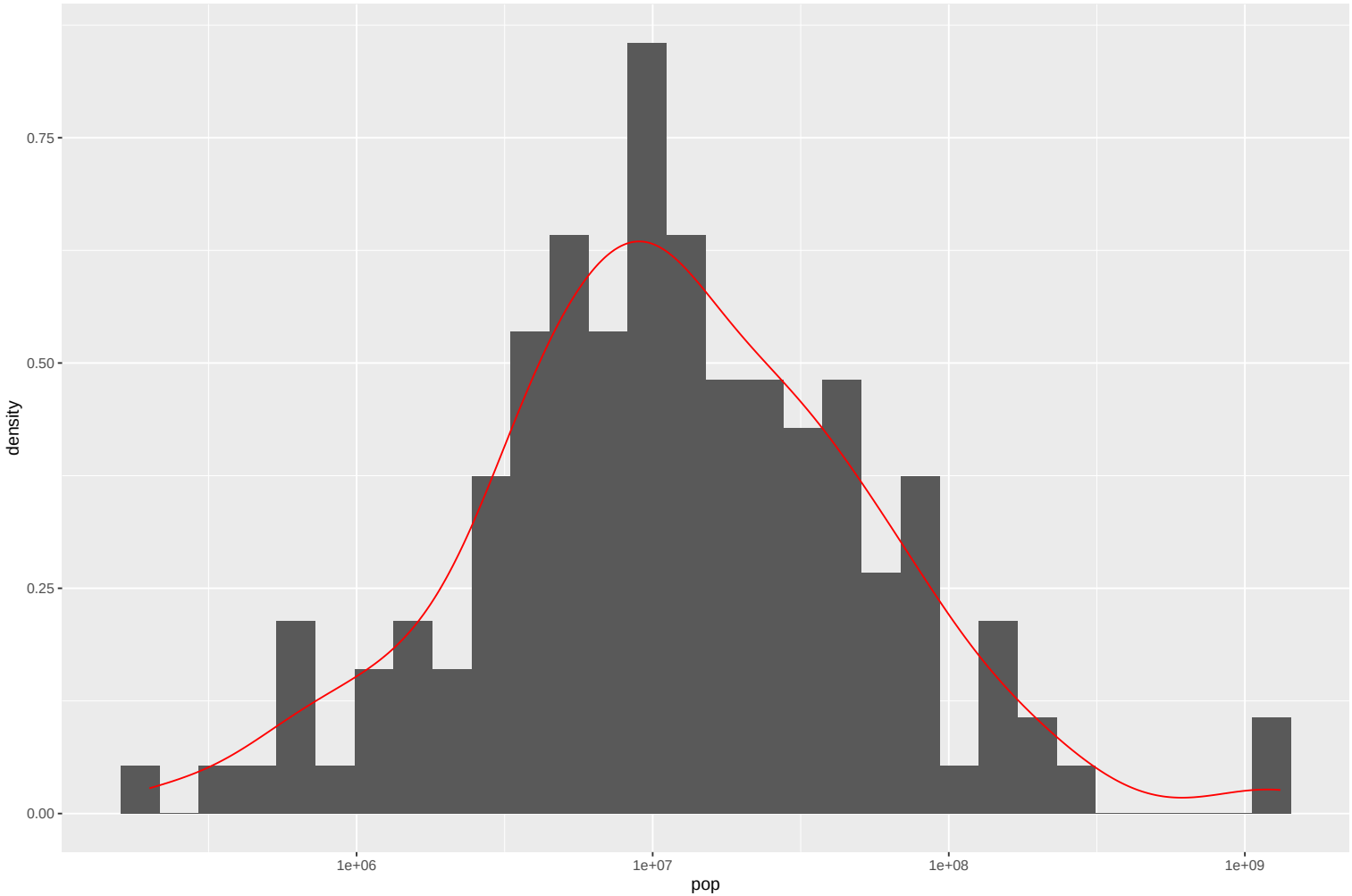
```
gapminder %>% filter(year==2007) %>% ggplot(aes(x=pop)) +
    geom_density() + scale_x_log10()
```



Kdensity+Histogram

```
gapminder %>%
    filter(year==2007) %>%
    ggplot(aes(x=pop)) +
    geom_histogram(aes(y = stat(density)))+
    geom_density(col="red") +
    scale_x_log10()
```

Kdensity+Histogram

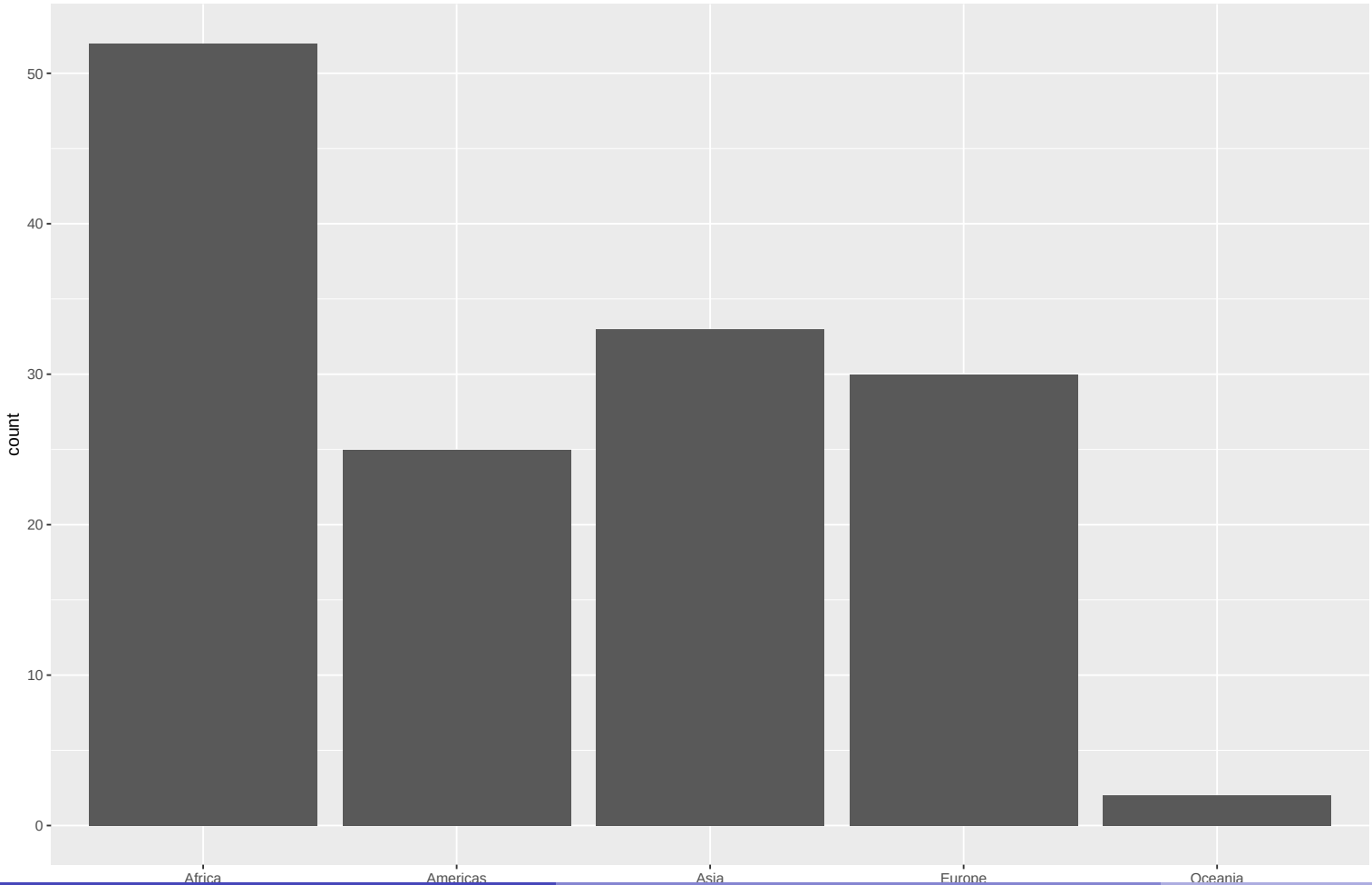


Bar

- `geom_bar()` and `geom_col()`.
- `geom_bar()` makes the height of the bar proportional to the number of cases in each group. It uses `stat_count()` by default: it counts the number of cases at each x position.
- If you want the heights of the bars to represent values in the data, use `geom_col()` instead. It uses stat_identity(): it leaves the data as is.

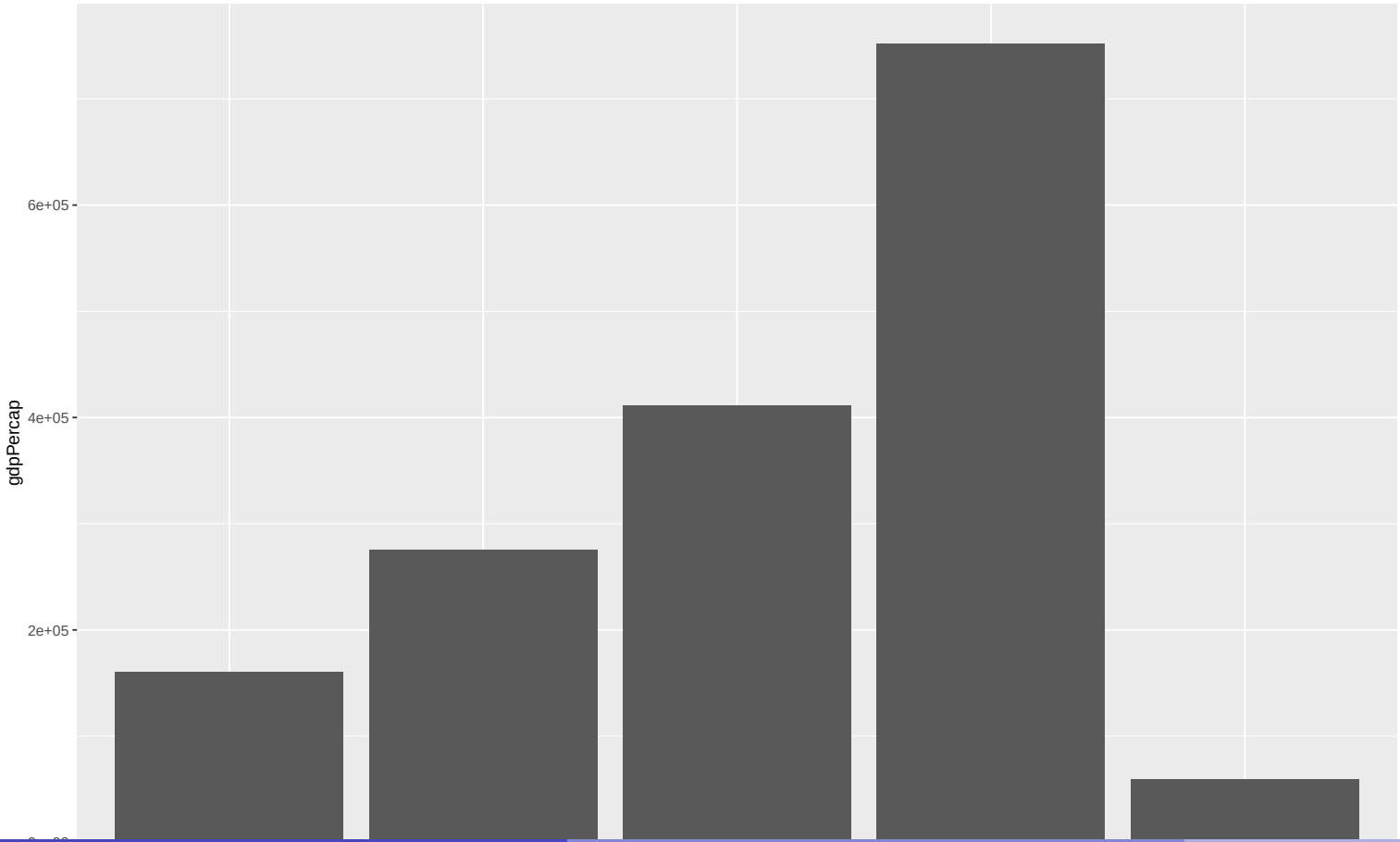
Bar

```
gapminder %>%
    filter(year==2007) %>%
    ggplot(aes(x=continent)) + geom_bar()
```



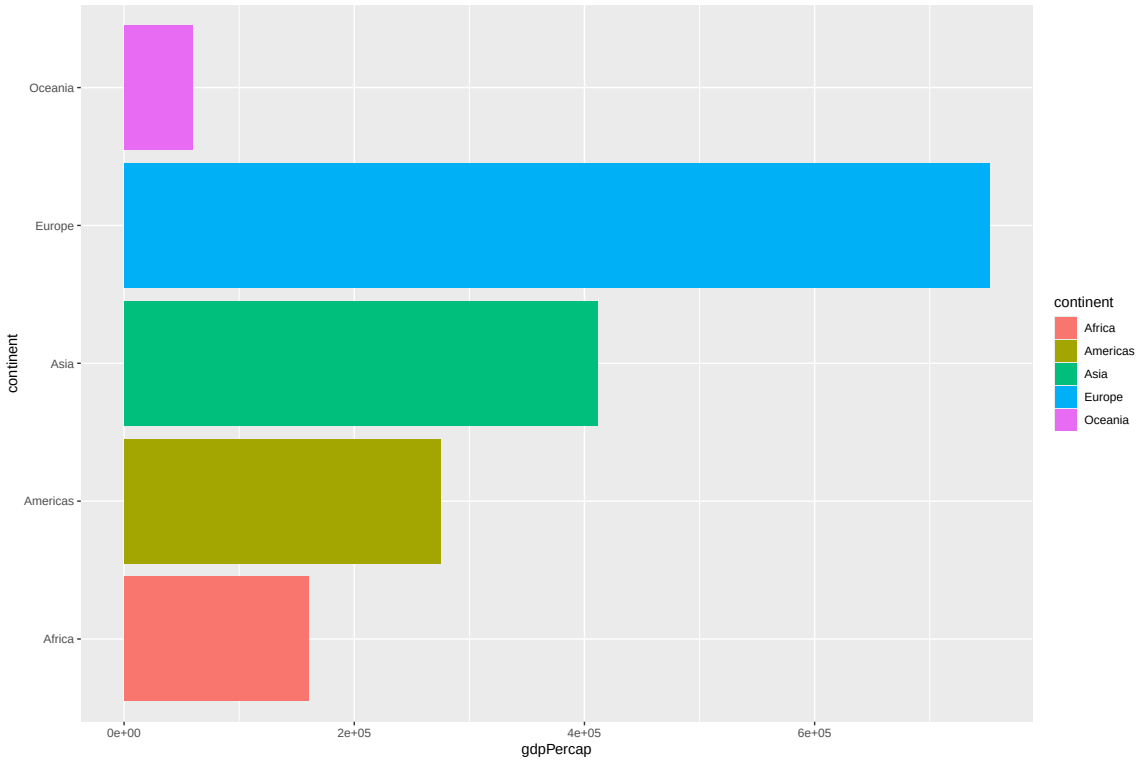
Bar

```
gapminder %>%
    filter(year==2007) %>%
    ggplot(aes(y=gdpPercap,x=continent)) +
    geom_col()
```



Bar

```
gapminder %>%
    filter(year==2007) %>%
    ggplot(aes(y=gdpPercap,x=continent,fill=continent)) +
    geom_col() + coord_flip()
```



Section 8

Introduction to Making Tables

Library Packages

```
library(tidyverse)
library(magrittr)
library(haven)
library(ggplot2)
library(AER)# package of Applied Econometrics in R
library(stargazer)
```

Reading the Data

```
ca <- read_dta("Data/caschool.dta")
```

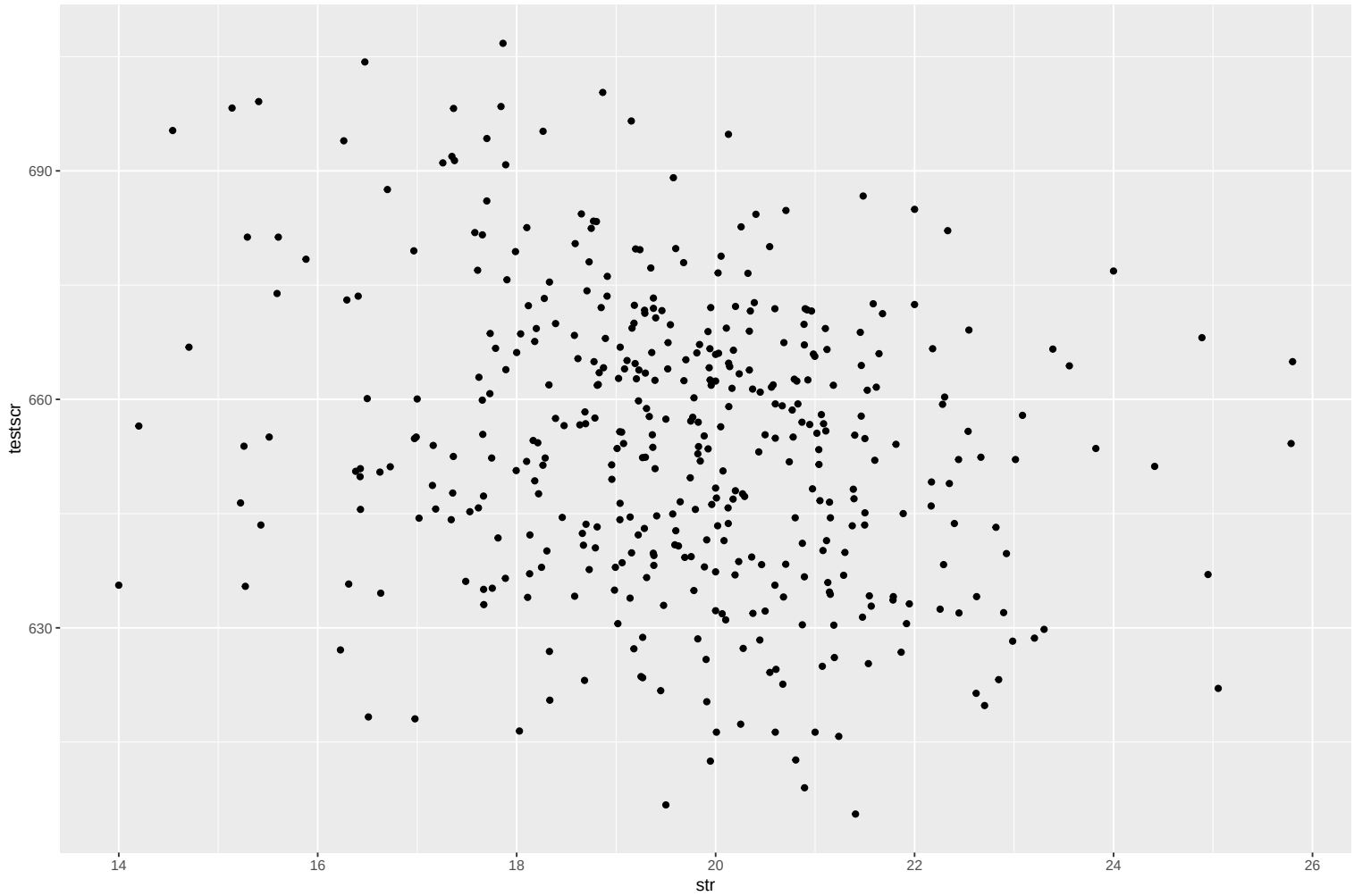
Overlook the Data

```
glimpse(ca,width = 90,size='tiny')
```

```
## Rows: 420
## Columns: 18
## $ observat <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,
## $ dist_cod <dbl> 75119, 61499, 61549, 61457, 61523, 62042,
## $ county   <chr> "Alameda", "Butte", "Butte", "Butte", "But
## $ district <chr> "Sunol Glen Unified", "Manzanita Elementar
## $ gr_span  <chr> "KK-08", "KK-08", "KK-08", "KK-08", "KK-08
## $ enrl_tot <dbl> 195, 240, 1550, 243, 1335, 137, 195, 888,
## $ teachers <dbl> 10.90, 11.15, 82.90, 14.00, 71.50, 6.40, 1
## $ calw_pct <dbl> 0.5102, 15.4167, 55.0323, 36.4754, 33.1086
## $ meal_pct <dbl> 2.0408, 47.9167, 76.3226, 77.0492, 78.4270
## $ computer <dbl> 67, 101, 169, 85, 171, 25, 28, 66, 35, 0,
## $ testscr  <dbl> 690.80, 661.20, 643.60, 647.70, 640.85, 60
## $ comp_stu <dbl> 0.34358975, 0.42083332, 0.10903226, 0.3497
```

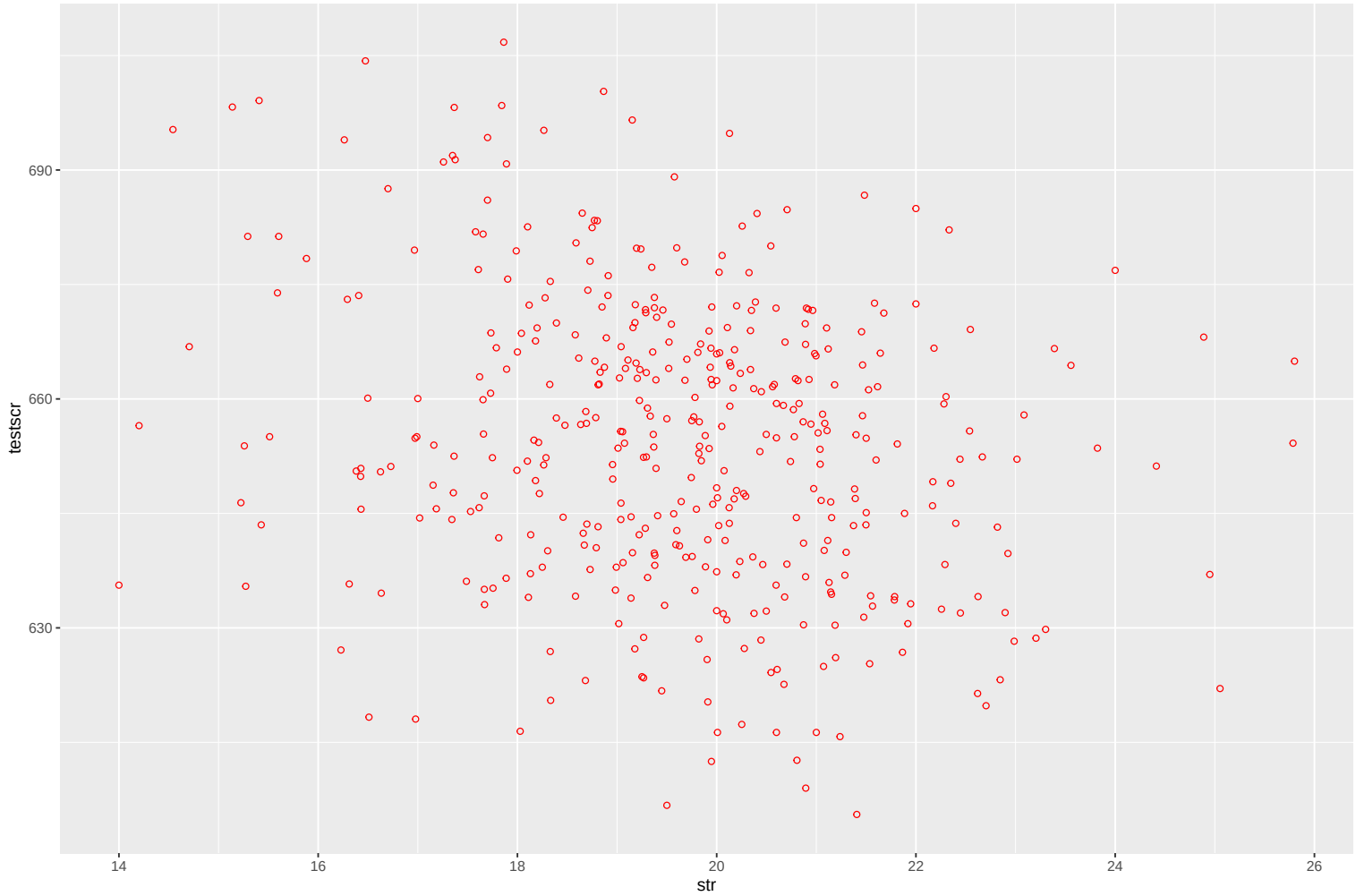
ggplot2 in action: Scatter

```
ggplot(data =ca,aes(x=str, y=testscr))+ geom_point()
```



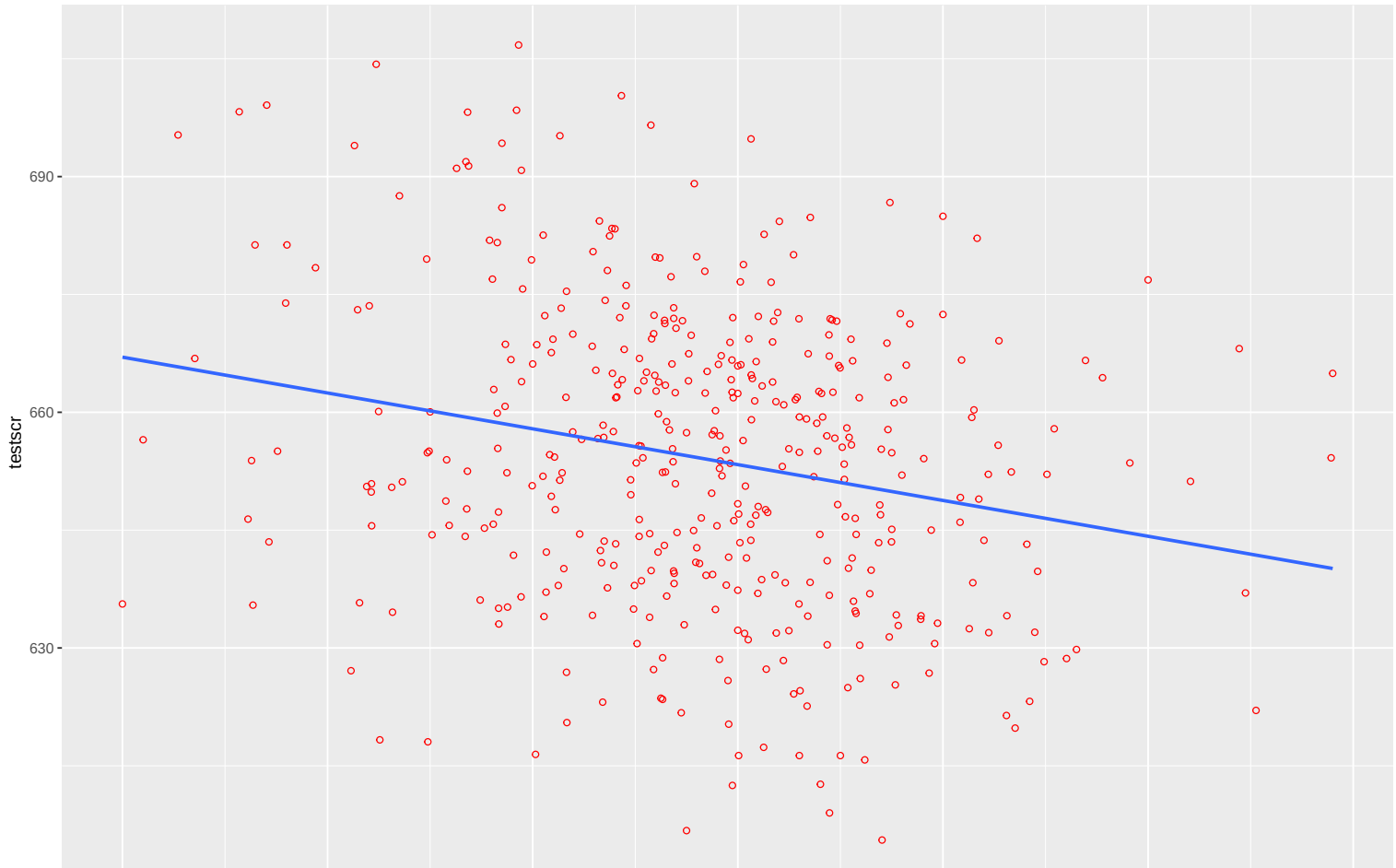
ggplot2 in action: Scatter

```
ggplot(data =ca,aes(x=str, y=testscr))+
  geom_point(shape=1,colour="red") # Use hollow circles and tu
```



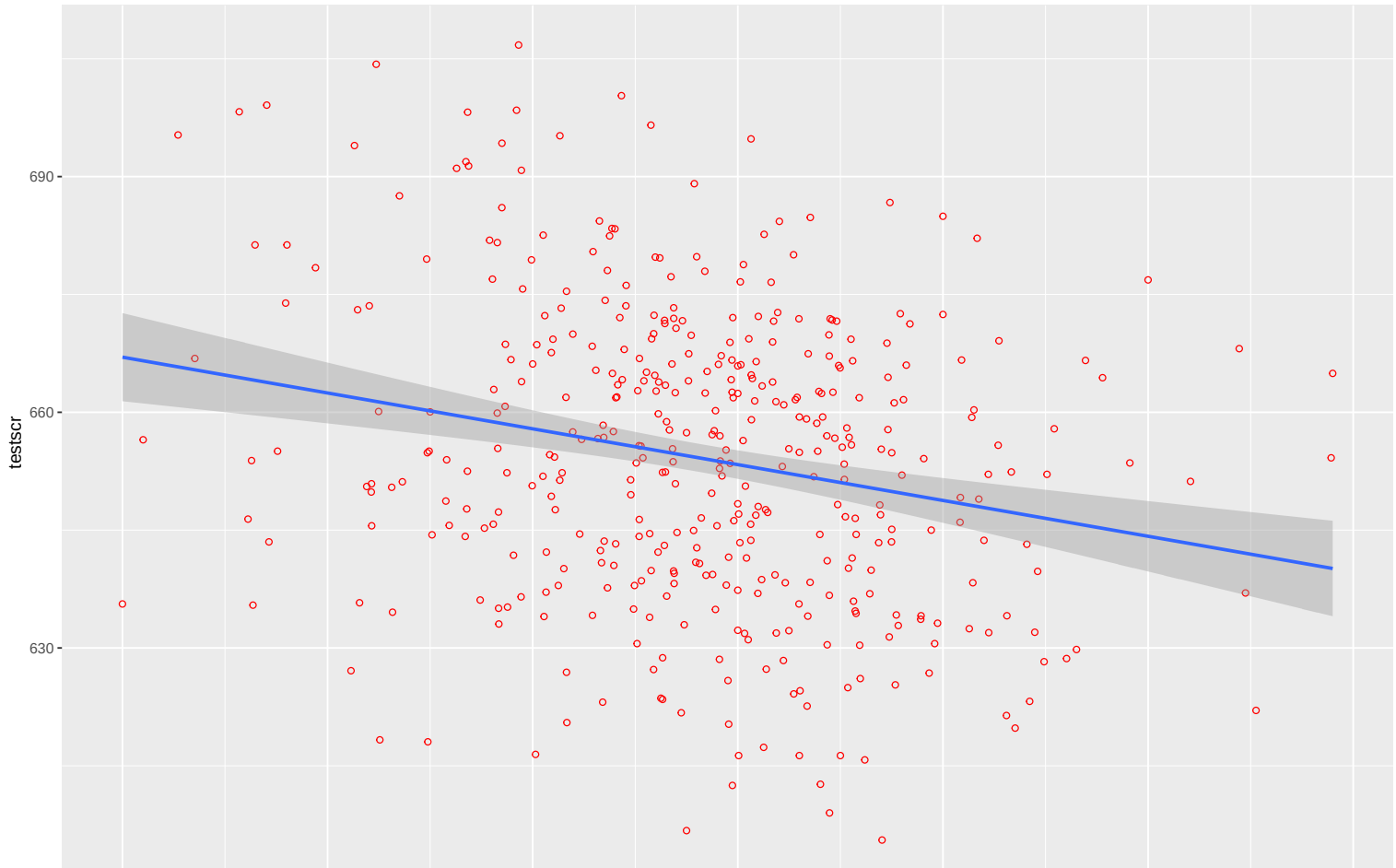
ggplot2 in action: Scatter

```
ggplot(data =ca,aes(x=str, y=testscr))+
  geom_point(shape=1,colour="red") +
  geom_smooth(method=lm,se= F)
```



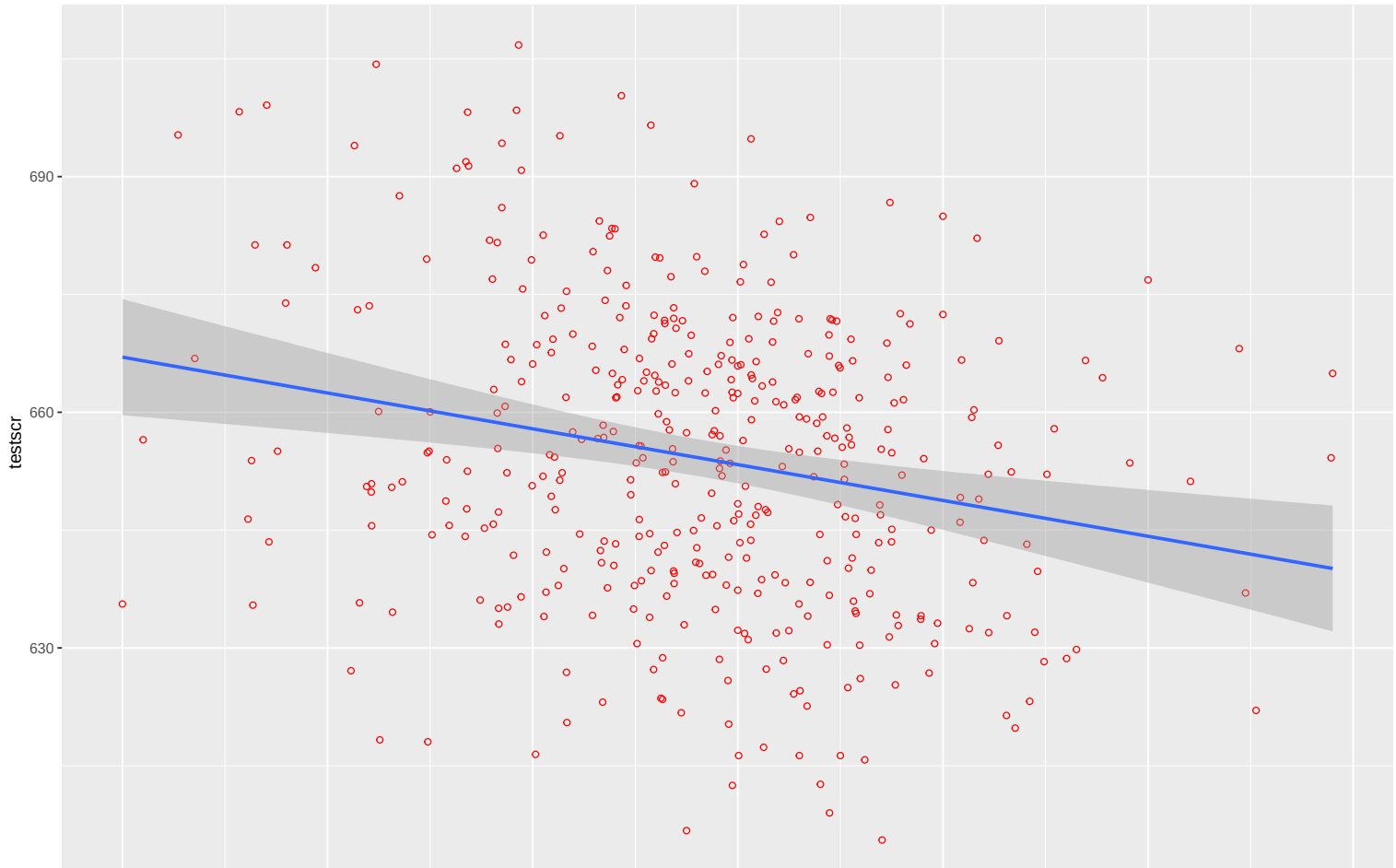
ggplot2 in action: Scatter

```
ggplot(data =ca,aes(x=str, y=testscr))+
  geom_point(shape=1,colour="red") + # Use hollow circles and
  geom_smooth(method=lm) # default with Confident Interval on
```



ggplot2 in action: Scatter

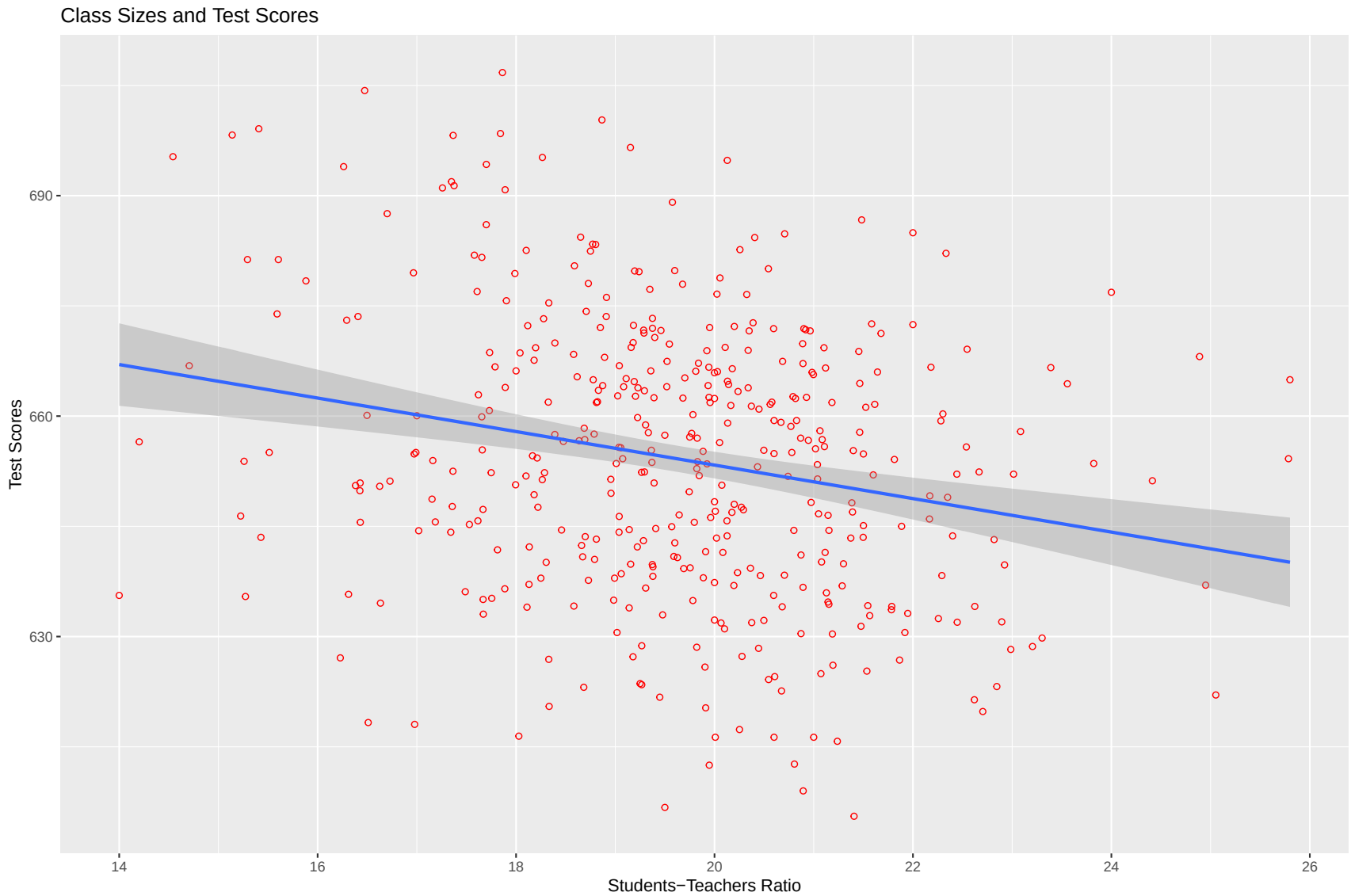
```
ggplot(data =ca,aes(x=str, y=testscr))+
  geom_point(shape=1,colour="red") + # Use hollow circles and
  geom_smooth(method=lm,level=0.99) # with Confident Interval
```



xlabel,ylabel and title

```
ggplot(data =ca,aes(x=str, y=testscr))+
  geom_point(shape=1,colour="red") + # Use hollow circles and
  geom_smooth(method=lm) + # with Confident Interval on 5% sig
  xlab("Students-Teachers Ratio") +
  ylab("Test Scores") +
  ggtitle("Class Sizes and Test Scores")
```

Scatter: xlabel,ylabel and title

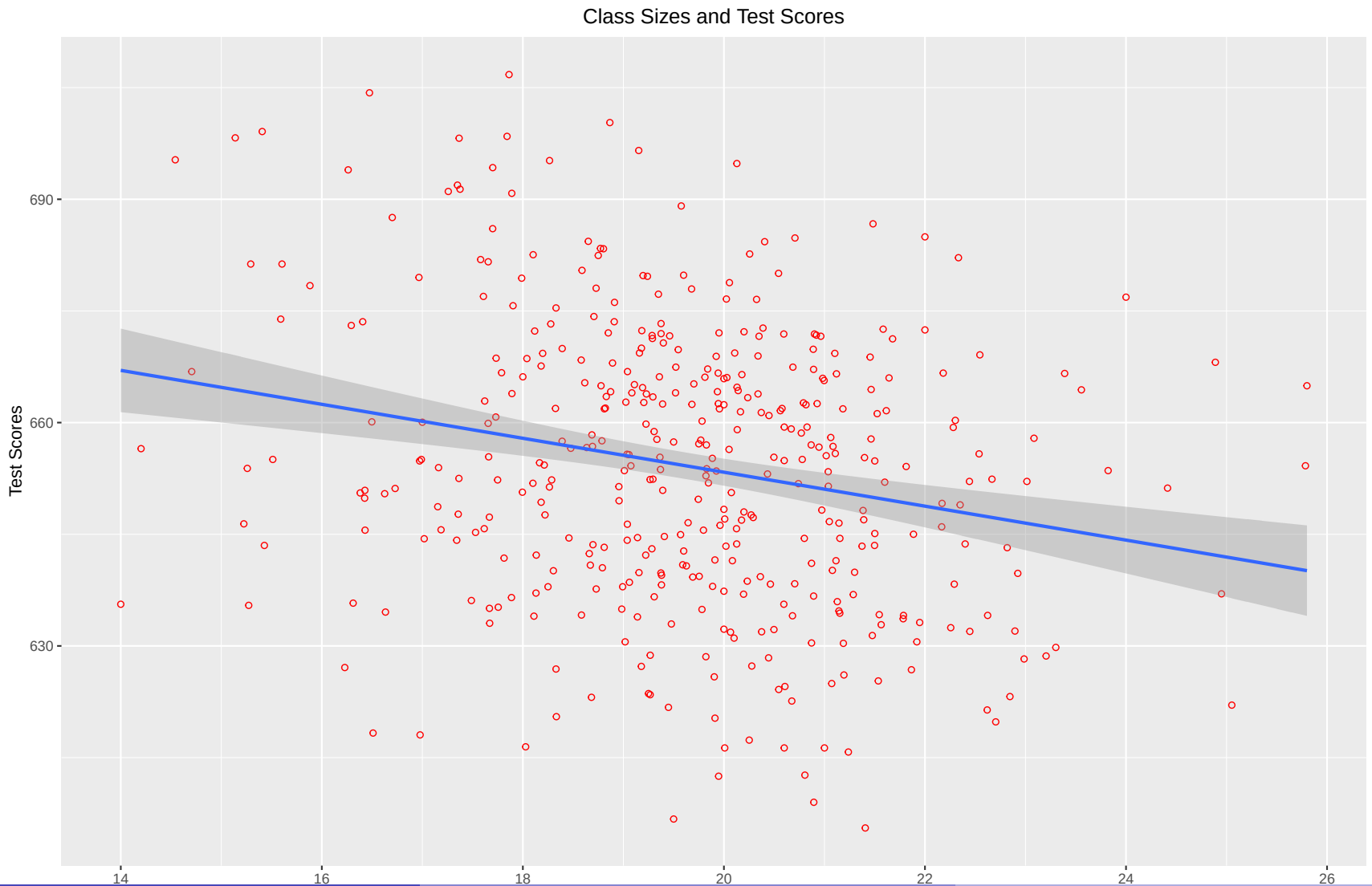


Scatter: title in center position

```
ggplot(data =ca,aes(x=str, y=testscr))+
  geom_point(shape=1,colour="red") + # Use hollow circles and
  geom_smooth(method=lm) + # with Confident Interval on 5% sig
  xlab("Students-Teachers Ratio") +
  ylab("Test Scores") +
  ggtitle("Class Sizes and Test Scores") +
  theme(plot.title = element_text(hjust = 0.5)) # title in cer
```

Scatter: title in center position

```
## `geom_smooth()` using formula = 'y ~ x'
```



Descriptive Table (1) in MS Word

```
ca_small <- ca %>%
  select(c(testscr,str,avginc,meal_pct,calw_pct,el_pct,
           expn_stu,comp_stu))
ca_data <-data.frame(ca_small) # data.frame for "stargazer" pa
```

Descriptive Table (2)

```
library(stargazer)
stargazer(ca_data,type = "text")
```

Descriptive Table (3): MS Word

```
##
## ==========================================================
## Statistic  N    Mean     St. Dev.     Min         Max
## ----------------------------------------------------------
## testscr   420  654.157    19.053    605.550    706.750
## str       420  19.640     1.892     14.000     25.800
## avginc    420  15.317     7.226      5.335     55.328
## meal_pct  420  44.705     27.123     0.000     100.000
## calw_pct  420  13.246     11.455     0.000     78.994
## el_pct    420  15.768     18.286     0.000     85.540
## expn_stu  420 5,312.408 633.937  3,926.070 7,711.507
## comp_stu  420   0.136     0.065      0.000      0.421
## ----------------------------------------------------------
```

Descriptive Table (4): Specific Statistics

```
stargazer(ca_data,type = "text",
          summary.stat = c("n", "mean", "sd","max","min"),
          digits = 2,out = "table1.txt")
```

Descriptive Table (5): Specific Statistics

```
##
## ==========================================================
## Statistic  N     Mean    St. Dev.    Max        Min
## ----------------------------------------------------------
## testscr    420   654.16    19.05     706.75    605.55
## str        420   19.64      1.89     25.80     14.00
## avginc     420   15.32      7.23     55.33      5.34
## meal_pct   420   44.71     27.12     100.00     0.00
## calw_pct   420   13.25     11.45     78.99      0.00
## el_pct     420   15.77     18.29     85.54      0.00
## expn_stu   420 5,312.41   633.94   7,711.51 3,926.07
## comp_stu   420    0.14      0.06      0.42      0.00
## ----------------------------------------------------------
```

Descriptive Table (6): Labelling Variables

```
stargazer(ca_data,type = "text",
    summary.stat = c("n", "mean", "sd","max","min"),
    covariate.labels=c("test score","student-teacher ratio"),
    digits = 2,out = "table1.txt")
```

Descriptive Table (7): Labelling Variables

```
##
## ==========================================================
## Statistic                  N     Mean    St. Dev.    Max       Min
## ----------------------------------------------------------
## test score                 420   654.16    19.05     706.75    605.5
## student-teacher ratio      420   19.64      1.89     25.80     14.00
## avginc                     420   15.32      7.23     55.33      5.34
## meal_pct                   420   44.71     27.12     100.00     0.00
## calw_pct                   420   13.25     11.45     78.99      0.00
## el_pct                     420   15.77     18.29     85.54      0.00
## expn_stu                   420 5,312.41   633.94   7,711.51 3,926.
## comp_stu                   420    0.14      0.06      0.42      0.00
## ----------------------------------------------------------
```

Descriptive Table (8): Title and Notes

```
stargazer(ca_data,type = "text",
    summary.stat = c("n", "mean", "sd","max","min"),
    covariate.labels=c("test score","student-teacher ratio"),
    digits = 2,
    title = "Descriptive statistics",
    notes = "Something you want to say about the table",
    notes.append = T)
```

Descriptive Table (9): Title and Notes

```
##
## Descriptive statistics
## ==========================================================
## Statistic                  N     Mean    St. Dev.    Max       Min
## ----------------------------------------------------------
## test score                 420   654.16    19.05     706.75    605.5
## student-teacher ratio      420   19.64      1.89     25.80     14.00
## average income             420   15.32      7.23     55.33      5.34
## meal_pct                   420   44.71     27.12     100.00     0.00
## calw_pct                   420   13.25     11.45     78.99      0.00
## el_pct                     420   15.77     18.29     85.54      0.00
## expn_stu                   420 5,312.41   633.94   7,711.51 3,926.
## comp_stu                   420    0.14      0.06      0.42      0.00
## ----------------------------------------------------------
## Something you want to say about the table
```

Descriptive Table in LaTeX (1)

```
stargazer(ca_data,type = "latex")
```

- chunk options: `results='asis'`

```
```{r echo=FALSE, message=FALSE, warning=FALSE, results='asis',eval=FALSE}
stargazer(ca_data,type = "latex",
          no.space=TRUE,header = FALSE,column.sep.width = "5pt",
          digits = 2,font.size = "small")
```
```

Descriptive Table in LaTeX (2)

```
stargazer(ca_data,type = "latex")
```

Descriptive Table in LaTeX (2)

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Mon, Mar 17, 2025 - 22:52:04

Statistic	N	Mean	St. Dev.	Min	Max
testscr	420	654.157	19.053	605.550	706.750
str	420	19.640	1.892	14.000	25.800
avginc	420	15.317	7.226	5.335	55.328
meal_pct	420	44.705	27.123	0.000	100.000
calw_pct	420	13.246	11.455	0.000	78.994
el_pct	420	15.768	18.286	0.000	85.540
expn_stu	420	5,312.408	633.937	3,926.070	7,711.507
comp_stu	420	0.136	0.065	0.000	0.421

Descriptive Table in LaTeX (3)

```
stargazer(ca_data,type = "latex",
          no.space=TRUE,header = FALSE,
          column.sep.width = "5pt",
          digits = 2,font.size = "small")
```

Descriptive Table in LaTeX (3)

Statistic	N	Mean	St. Dev.	Min	Max
testscr	420	654.16	19.05	605.55	706.75
str	420	19.64	1.89	14.00	25.80
avginc	420	15.32	7.23	5.34	55.33
meal_pct	420	44.71	27.12	0.00	100.00
calw_pct	420	13.25	11.45	0.00	78.99
el_pct	420	15.77	18.29	0.00	85.54
expn_stu	420	5,312.41	633.94	3,926.07	7,711.51
comp_stu	420	0.14	0.06	0.00	0.42

Descriptive Table in LaTeX (4): Specific Statistics

```
stargazer(ca_data,type = "latex",
          summary.stat = c("n", "median", "sd","max","min"),
          no.space=TRUE,header = FALSE,
          column.sep.width = "5pt",
          digits = 2,font.size = "small")
```

Descriptive Table in LaTeX (4): Specific Statistics

Statistic	N	Median	St. Dev.	Max	Min
testscr	420	654.45	19.05	706.75	605.55
str	420	19.72	1.89	25.80	14.00
avginc	420	13.73	7.23	55.33	5.34
meal_pct	420	41.75	27.12	100.00	0.00
calw_pct	420	10.52	11.45	78.99	0.00
el_pct	420	8.78	18.29	85.54	0.00
expn_stu	420	5,214.52	633.94	7,711.51	3,926.07
comp_stu	420	0.13	0.06	0.42	0.00

Descriptive Table in LaTeX (5): Labels

```
stargazer(ca_data,type = "latex",
  summary.stat = c("n", "mean", "sd","max","min"),
  covariate.labels=c("test score","student-teacher ratio",
                     "average income","free lunch",
                     "low-income program",
                     "english-learners",
                     "expenditure per student",
                     "computers per student"),
          no.space=TRUE,header = FALSE,
          column.sep.width = "5pt",
          digits = 2,font.size = "small")
```

Descriptive Table in LaTeX (5): Labels

Statistic	N	Mean	St. Dev.	Max	Min
test score	420	654.16	19.05	706.75	605.55
student-teacher ratio	420	19.64	1.89	25.80	14.00
average income	420	15.32	7.23	55.33	5.34
free lunch	420	44.71	27.12	100.00	0.00
low-income program	420	13.25	11.45	78.99	0.00
english-learners	420	15.77	18.29	85.54	0.00
expenditure per student	420	5,312.41	633.94	7,711.51	3,926.07
computers per student	420	0.14	0.06	0.42	0.00

Descriptive Table in LaTeX (6): Title and Notes

```
stargazer(ca_data,type = "latex",
          summary.stat = c("n", "mean", "sd","max","min"),
          covariate.labels=c("test score",
                             "student-teacher ratio",
                             "average income","free lunch",
                             "low-income program",
                             "english-learners",
                             "expenditure per student",
                             "computers per student"),
          no.space=TRUE,header = FALSE,
          column.sep.width = "5pt",
          digits = 2,font.size = "small",
          title = "Table1: Descriptive statistics",
          notes = "Something you want to say about the table",
          notes.append = T)
```

Descriptive Table in LaTeX (6): Title and Notes

Table1: Descriptive statistics

Statistic	N	Mean	St. Dev.	Max	Min
test score	420	654.16	19.05	706.75	605.55
student-teacher ratio	420	19.64	1.89	25.80	14.00
average income	420	15.32	7.23	55.33	5.34
free lunch	420	44.71	27.12	100.00	0.00
low-income program	420	13.25	11.45	78.99	0.00
english-learners	420	15.77	18.29	85.54	0.00
expenditure per student	420	5,312.41	633.94	7,711.51	3,926.07
computers per student	420	0.14	0.06	0.42	0.00

Something you want to say about the table

Descriptive Table in LaTeX (7): 中文表示

```
stargazer(ca_data,type = "latex",
          summary.stat = c("n", "mean", "sd","max","min"),
          covariate.labels=c(" 分数"," 生师比"," 家庭人均收入",
                             " 免费午餐比例"," 低收入项目比例"," 外来移
                             " 生均花费"," 生均计算机数量"),
          no.space=TRUE,header = FALSE,column.sep.width = "5pt
          digits = 2,font.size = "small",
          title = " 表 1: 各变量描述性统计",
          notes = "1) 表格注意事项",
          notes.append = T)
```

Descriptive Table in LaTeX (7): 中文表示

表 1: 各变量描述性统计

Statistic	N	Mean	St. Dev.	Max	Min
分数	420	654.16	19.05	706.75	605.55
生师比	420	19.64	1.89	25.80	14.00
家庭人均收入	420	15.32	7.23	55.33	5.34
免费午餐比例	420	44.71	27.12	100.00	0.00
低收入项目比例	420	13.25	11.45	78.99	0.00
外来移民比例	420	15.77	18.29	85.54	0.00
生均花费	420	5,312.41	633.94	7,711.51	3,926.07
生均计算机数量	420	0.14	0.06	0.42	0.00

1) 表格注意事项

Reference

1. Grolemund & Wickham(2017), R for Data Science
2. DataCamp(2018), Introduction to Tidyverse
3. John Sullivan(2019), Data Cleaning with R and the Tidyverse: Detecting Missing Values