

PSYCH 213 Cheat Sheet

1 Operators

<code>x <- 5</code>	assign 5 to x
<code>x = 5</code>	also assigns, but <- is preferred
<code>x > y</code>	greater than
<code>x < y</code>	less than
<code>x >= y</code>	greater or equal
<code>x <= y</code>	less or equal
<code>x == y</code>	equal
<code>x != y</code>	not equal
<code>!x</code>	NOT x
<code>x y</code>	x OR y
<code>x & y</code>	x AND y
<code>x > f()</code>	pipe: send x into a function f()

2 Vectors

Create a vector

```
nums <- c(10, 20, 30, 40)
words <- c("a", "b", "c")
logical <- c(TRUE, FALSE, T, F)
```

Indexing a vector

```
nums[1]      # first value (10)
nums[2:3]    # values 2 through 3 (20, 30)
nums[-2]     # all but the 2nd value
nums[c(1,4)] # first and fourth values (10, 40)
```

3 Useful Functions

<code>length(x)</code>	number of values in x
<code>nrow(df)</code>	number of rows in a data frame
<code>sum(x)</code>	sum of numbers
<code>mean(x)</code>	average
<code>median(x)</code>	middle value
<code>sd(x)</code>	standard deviation
<code>sort(x)</code>	sort smallest to largest
<code>is.na(x)</code>	check for missing values
<code>paste0("a", "b")</code>	combine text
<code>table(x)</code>	count values
<code>rep(x, times = n)</code>	repeat values in a vector
<code>seq(from, to, by)</code>	generate a sequence of numbers

4 Working With Data

Read data

Base R:

```
data <- read.csv("mydata.csv")
```

Tidyverse:

```
library(tidyverse)
data <- read_csv("mydata.csv")
```

4.1 Make a data frame

```
df <- data.frame(
  name=c("A", "B", "C"),
  score=c(10, 20, 30)
)
```

Subset data frames with indexing

```
df$score      # full column
df[2, ]       # row 2
df[, 2]       # column 2
df[2, 2]      # row 2, column 2
```

Subset data frames with functions

Base R:

```
subset(df, score > 15 & name == "B")
```

Tidyverse:

```
filter(df, score > 15 & name == "B")
```

5 Plots (with ggplot2)

```
library(tidyverse)

# Histogram
ggplot(df, aes(x = score)) +
  geom_histogram()

# Barplot
ggplot(df, aes(x = name, y = score)) +
  geom_bar(stat = "identity")

# Boxplot
ggplot(df, aes(x = name, y = score)) +
  geom_boxplot()
```

```
# Scatterplot
ggplot(df, aes(x = name, y = score)) +
  geom_point()
```

6 Aggregating with Pipes and the Tidyverse

```
library(tidyverse)

msleep |>
  group_by(vore) |>
  summarise(
    n = length(brainwt),
    m = mean(brainwt, na.rm = TRUE),
    s = sd(brainwt, na.rm = TRUE)
  )
```

7 Probability and Quantile Functions

<code>pnorm(q, mean = 0, sd = 1)</code>	probability below z-score (area to left)
<code>qnorm(p, mean = 0, sd = 1)</code>	z-score (quantile) for given probability
<code>pt(q, df)</code>	probability below t-score
<code>qt(p, df)</code>	t-score value for given probability
<code>pchisq(q, df)</code>	probability below chi-square value
<code>qchisq(p, df)</code>	chi-square value for given probability

The argument `lower.tail = FALSE` gives upper-tail probabilities or values.

8 T-testing

One-sample

```
# Compares single mean to null hypothesized mean
t.test(x,
  alternative = c("two.sided", "less", "greater"), # pick only one
  mu = 0
)
```

Paired Two-Sample

```
# Compares the difference between two means for paired/dependent conditions.
t.test(x, y,
  alternative = c("two.sided", "less", "greater"), # pick only one
  mu = 0,
  paired = TRUE
)
```

Independent Two-Sample

```
# Compares the difference between two means for independent conditions.
t.test(x, y,
  alternative = c("two.sided", "less", "greater"), # pick only one
  mu = 0,
  var.equal = TRUE
)
```

9 Linear Regression

Simple Linear Regression

```
# y = b0 + b1*x
model <- lm(y ~ x, data = df)
summary(model)
confint(model)
```

10 Goodness-of-Fit and Chi-Square Testing

Goodness-of-Fit Test

```
# Tests whether observed frequencies match expected proportions
observed <- c(20, 50, 30)
expected <- c(0.25, 0.5, 0.25) # proportions sum to 1
chisq.test(x = observed, p = expected)
```

One-Way (Pearson) Chi-Square Test of Independence

```
# Tests whether two categorical variables are independent
tbl <- table(df$var_A, df$var_B)
chisq.test(tbl)
```

Remember: R is case-sensitive!